Anti-Forensic Techniques Used By Lazarus Group

By AFIRST.SH ⁝⁝ 2/23/2023

Since approximately a year ago, the Lazarus group’s malware has been discovered in various Korean companies related to national defense, satellites, software, and media press. The AhnLab ASEC analysis team has been continuously tracking the Lazarus threat group’s activities and other related TTPs.

Among the recent cases, this post aims to share the anti-forensic traces and details found in the systems that were infiltrated by the Lazarus group.

Overview

Definition of Anti-Forensics

Anti-forensics refers to the tampering of evidence in an attempt to mitigate the effectiveness of a forensics investigation at a crime scene. From a breaching point of view, anti-forensics generally have the following objectives.

- Detection evasion and obstruction of information collection
- Increase the analysis time of digital forensic analysts
- Disable or cause the malfunction of digital forensic tools
- Block, bypass or delete logs to hide traces of access or execution of tools

The Lazarus group carried out anti-forensics to conceal their malicious acts.

Anti-Forensic Techniques

While there are various standards on the classification of anti-forensic techniques, this post will use the most widely received anti-forensic classification proposed by Dr. Marcus Roger to distinguish and analyze concealment measures taken by the Lazarus group.

Dr. Marcus Roger classified anti-forensic techniques which hinder forensic analysis into 5 main categories: data hiding, artifact wiping, trail obfuscation, attacks against computer forensics, and physical.

Looking at the 5 categories above, the Lazarus group utilized data hiding, artifact wiping, and trail obfuscation, a total of 3 techniques.

Data Hiding

Data hiding refers to the method of data concealment that renders their detection difficult. Major examples include data obfuscation, encryption, steganography, and hiding data in non-allocated areas.

Encryption

The Lazarus group distinguished and used their malware in 3 parts. The loader, executable file, and configuration file. The major features of each file are as follows.

- Loader: Decrypts encrypted PE files and loads them onto the memory
- Encrypted PE: A malware that runs on the loader memory and decrypts encrypted configuration files to communicate with the C2 address.
- Encrypted Config: An encrypted configuration file that contains C2 information.

The Lazarus group transmits the configuration file that has the C2 information and the PE file that communicates with the C2 in encrypted forms to evade detection by security products. The encrypted files operate after being decrypted onto the memory by the loader file. They then receive additional files from the C2 and perform malicious acts.

Figure. Backdoor operation process

Other Forms of Data Hiding

The Lazarus group either used a system folder as a hiding place or imitated the name of a normal file to hide their malware. The default system folders are where their malware hiding is mainly done. The malware is hidden by either creating a similar folder within the system or by disguising the malware as a normal file within a system file that's hidden by default.

- C:\ProgramData\
- C:\ProgramData\Microsoft\
- C:\Windows\System32\

The C:\ProgramData folder is a default system folder which is hidden by default. A folder with a name similar to the default folder (MicrosoftPackages) is created inside this folder as a malware hiding place or the malware is disguised as a similar file inside the default folder.

Figure. Similar folder created to use as a malware hiding place

Figure. Hiding malware by imitating default folder names

Artifact Wiping

Artifact wiping refers to the task of permanently deleting specific files or the whole file system. Not only does it involve file deletion, but expert tools can also be used to erase all traces of use. For example, the Disk Clean-up utility, file deletion, and disk demagnetization are all included in artifact wiping.

File Wiping

Excluding the backdoor malware, the Lazarus group deleted the malware and the artifacts that occurred while the malicious behavior was being performed. In the malware’s case, its data was overwritten and its filename was changed before being deleted.

The original file content can no longer be seen if the data section is overwritten during file deletion since this makes data recovery through methods such as file restoration and data carving difficult.

Figure. Malware deletion log confirmed

in USNJrnl

The Lazarus group also deleted artifacts related to their malware execution at the same time. For example, the prefetch files, which are artifacts related to application execution, were collectively deleted to remove traces of the malware being executed.

Figure. Log showing the collective deletion of prefetch files

Trail Obfuscation

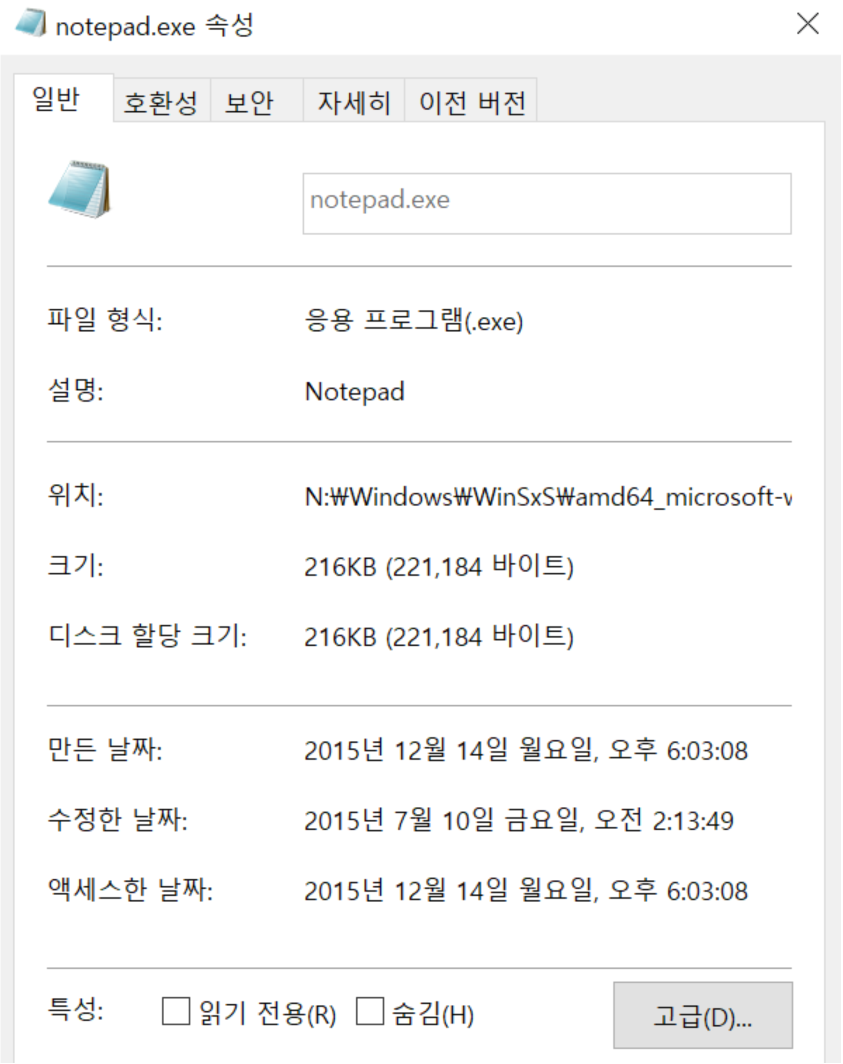
Trail obfuscation refers to the task of confusing the forensic process to hide malicious behavior. Modification/Deletion of logs, spoofing, inserting incorrect information, and backbone hopping can be considered examples of ways to interfere with analysis or confuse analysts.

Timestamp Changes

During the artifact wiping process, the Lazarus group would delete almost all the files and logs they used; however, there is a single malware strain that they would leave behind in the system, the backdoor malware. Backdoors characteristically must persist for long periods of time on the target system without being detected, so the Lazarus group modifies the malware creation timestamp information to hide the backdoor.

It appears that the biggest reason they are altering the time information is to evade timeline analysis. Timeline analysis refers to the analyzing technique that track the files that were created/modified/deleted/accessed around the time of an incident. When the timestamp information is altered, logs involving the malware are omitted from the analysis process, confusing the analysis direction, or leading to incorrect conclusions.

The timestamp of the malware that was found in a system was found to perfectly match the timestamp of some other file on the same system.



notepad.exe 속성

notepad.exe

파일 형식: 응용 프로그램(.exe)

설명: Notepad

위치: N:₩Windows₩WinSxS₩amd64_microsoft-v

크기: 216KB (221,184 바이트)

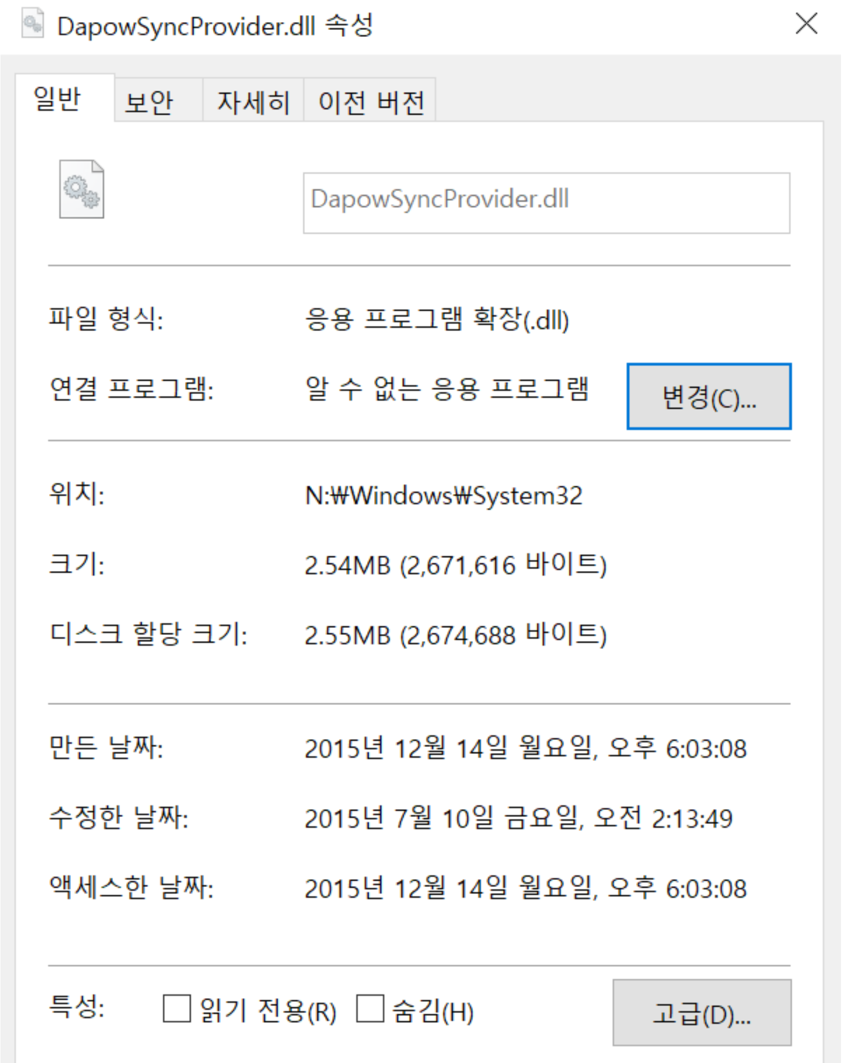
디스크 할당 크기: 216KB (221,184 바이트)

만든 날짜: 2015년 12월 14일 월요일, 오후 6:03:08

수정한 날짜: 2015년 7월 10일 금요일, 오전 2:13:49

액세스한 날짜: 2015년 12월 14일 월요일, 오후 6:03:08

특성: ☐ 읽기 전용(R) ☐ 숨김(H)



DapowSyncProvider.dll 속성

DapowSyncProvider.dll

파일 형식: 응용 프로그램 확장(.dll)

연결 프로그램: 알 수 없는 응용 프로그램

위치: N:₩Windows₩System32

크기: 2.54MB (2,671,616 바이트)

디스크 할당 크기: 2.55MB (2,674,688 바이트)

만든 날짜: 2015년 12월 14일 월요일, 오후 6:03:08

수정한 날짜: 2015년 7월 10일 금요일, 오전 2:13:49

액세스한 날짜: 2015년 12월 14일 월요일, 오후 6:03:08

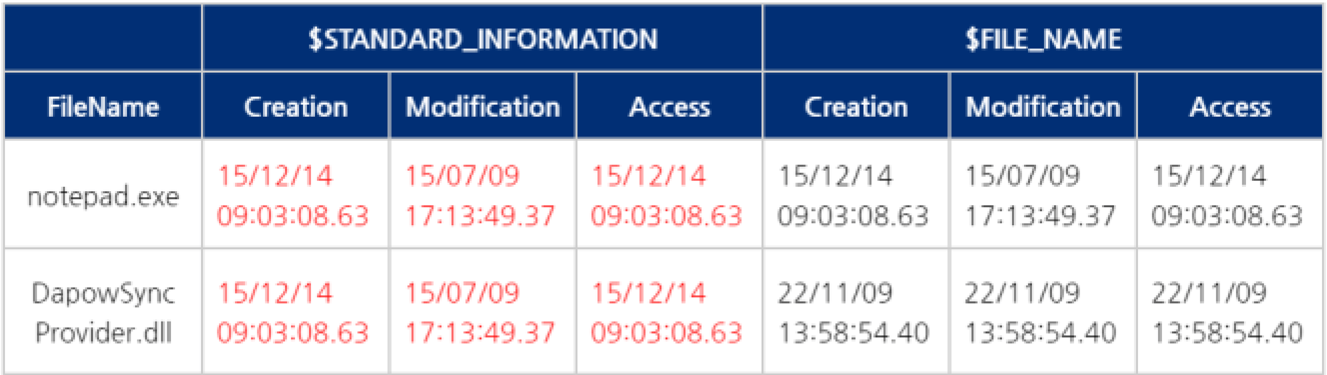
특성: ☐ 읽기 전용(R) ☐ 숨김(H)

Figure. Timestamp comparison of a default system file (notepad.exe) and malware (DapowSyncProvider.dll)

The $STANDARD_INFORMATION property and the $FILE_NAME property respectively have 4 timestamps in the Windows file system (NTFS): the file creation, modification, access, and entry modification timestamps. The timestamps displayed on the Windows file properties tab is the data from $STANDARD_INFORMATION.

- $STANDARD_INFORMATION: property that exists by default in all files that contains the basic information of files such as their timestamps, characteristics, owner ID, and security ID.
- $FILE_NAME: property that exists by default in all files that contains the filename and various other additional data of files as its purpose is to save the name of files.

$MFT was extracted and the timestamps were compared for a detailed timestamp analysis. As a result, the team discovered that the timestamp in the $STANDARD_INFORMATION property of a default system file (notepad.exe) and the malware (DapowSyncProvider.dll) matched perfectly.



	$STANDARD_INFORMATION			$FILE_NAME		
FileName	Creation	Modification	Access	Creation	Modification	Access
notepad.exe	15/12/14 09:03:08.63	15/07/09 17:13:49.37	15/12/14 09:03:08.63	15/12/14 09:03:08.63	15/07/09 17:13:49.37	15/12/14 09:03:08.63
DapowSync Provider.dll	15/12/14 09:03:08.63	15/07/09 17:13:49.37	15/12/14 09:03:08.63	22/11/09 13:58:54.40	22/11/09 13:58:54.40	22/11/09 13:58:54.40

Table. Timestamp comparison between malware and a default system file

The Lazarus group changed the malware's timestamp so that it matched and appeared like a default system file in order to evade timeline analysis.

The timestamp modification details of malware found in systems that have recently been infiltrated by the Lazarus group are as follows.

Figure. Modification of malware timestamps

With the organized information above, the characteristics of the Lazarus group's malware timestamp modification method can be summarized in the following ways.

- Not all malware have their timestamps modified.
- The timestamps are not set to arbitrary values, but rather copied over from default system files.
- Considering that there are systems with modified and unmodified timestamps within the same incident, it can be assumed that timestamp modification is optional.
- It appears that the copy target is selectable as the same malware would have the timestamps of different system files.

The team has confirmed that the timestamp modification technique done by the Lazarus group is also being used by other APT groups to hide their malware.

APT Group	Description
APT28	APT28 has performed timestomping on victim files.
APT29	APT29 modified timestamps of backdoors to match legitimate Windows files.
APT32	APT32 has used scheduled task raw XML with a backdated timestamp of June 2, 2016. The group has also set the creation time of the files dropped by the second stage of the exploit to match the creation time of kernel32.dll. Additionally, APT32 has used a random value to modify the timestamp of the file storing the clientID.
Chimera	Chimera has used a Windows version of the Linux touch command to modify the date and timestamp on DLLs.
Kimsuky	Kimsuky has manipulated timestamps for creation or compilation dates to defeat anti-forensics.

APT Group	Description
Lazarus	APT38 has modified data timestamps to mimic files that are in the same folder on a compromised host.Several Lazarus Group malware families use timestomping, including modifying the last write timestamp of a specified Registry key to a random date, as well as copying the timestamp for legitimate .exe files (such as calc.exe or mspaint.exe) to its dropped files.
Rocke	Rocke has changed the timestamp of certain files.
TEMP.Veles	TEMP.Veles used timestomping to modify the $STANDARD_INFORMATION attribute on tools

Table. Groups using the timestamp modification technique (MITRE ATT&CK)

Conclusion

During the Lazarus group's attack process, they would either encrypt their malware or disguise them with system file names to evade detection by security products. They also modified the timestamps of files to hinder analysis. Moreover, they displayed their meticulousness by overwriting their data before deleting their execution traces to obstruct data recovery.

While analyzing the anti-forensic techniques used by the Lazarus group, it was revealed that various other APT groups have also been using the same techniques to erase their attack traces.

When investigating and analyzing incidents, users must consider the possibility of the threat actor using anti-forensic techniques. Continuous research on tracking methods is required to ensure that malware can be traced even when anti-forensic techniques are applied.

[File Detection]

- Trojan/Win.LazarShell (2021.11.30)
- Trojan/BIN.Encoded (2021.12.15)
- Trojan/BIN.Encoded (2021.12.15)
- Trojan/Win.LazarLoader (2022.09.06)
- Data/BIN.EncPe (2022.09.06)
- Data/BIN.Encoded (2022.10.04)
- Backdoor/Win.Lazardoor (2022.09.06)
- Data/BIN.EncPe (2022.09.06)
- Data/BIN.Encoded (2022.10.04)
- Trojan/Win.LazarLoader (2022.09.06)
- Data/BIN.EncodedPE (2022.09.06)
- Trojan/Win.Lazardoor (2022.12.09)
- Data/BIN.Lazarus (2022.12.09)
- Trojan/Win.Lazardoor (2022.12.09)
- Data/BIN.Lazarus (2022.12.09)
- Data/BIN.Lazarus (2022.12.10)
- Trojan/Win.Lazardoor (2023.01.11)
- Data/BIN.EncodedPE (2023.01.12)

- Data/BIN.Encoded (2023.01.12)
- Trojan/Win.Lazardoor (2023.01.11)
- Data/BIN.EncodedPE (2023.01.12)
- Data/BIN.Encoded (2023.01.11)

[File MD5]

- B3E03A41CED8C8BAA56B8B78F1D55C22
- 1E7D604FADD7D481DFADB66B9313865D
- 7870DECBC7578DA1656D1D1FF992313C
- B457E8E9D92A1B31A4E2197037711783
- 1F1A3FE0A31BD0B17BC63967DE0CCC29
- C16A6178A4910C6F3263A01929F306B9
- 202A7EEC39951E1C0B1C9D0A2E24A4C4
- 1F1A3FE0A31BD0B17BC63967DE0CCC29
- 8543667917A318001D0E331AEAE3FB9B
- CA9B6B3BCE52D7F14BABDBA82345F5B1
- 97BC894205D696023395CBD844FA4E37
- C7256A0FBAB0F437C3AD4334AA5CDE06
- FC8B6C05963FD5285BCE6ED51862F125
- 27DB56964E7583E19643BF5C98FFFD52
- 61B3C9878B84706DB5F871B4808E739A
- 6EA4E4AB925A09E4C7A1E80BAE5B9584
- BD47942E9B6AD87EB5525040DB620756
- 67D306C163B38A06E98DA5711E14C5A7
- C09B062841E2C4D46C2E5270182D4272
- 747177AAD5AEF020B82C6AEABE5B174F
- E73EAB80B75887D4E8DD6DF33718E3A5
- BA741FA4C7B4BB97165644C799E29C99
- 064D696A93A3790BD3A1B8B76BAAEEF3