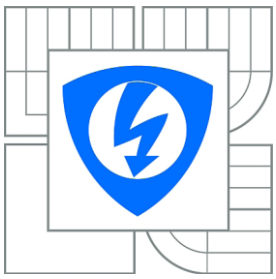
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

BEZPEČNOST OPERAČNÍCH SYSTÉMŮ PRO MOBILNÍ ZAŘÍZENÍ

SECURITY OF OPERATING SYSTEMS FOR MOBILE DEVICES

BAKALÁŘSKÁ PRÁCE

BACHELOR THESIS

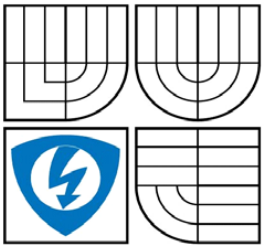
AUTOR PRÁCE — JAKUB KOLÁŘ

AUTHOR

VEDOUCÍ PRÁCE — Ing. MARTIN ROSENBERG

SUPERVISOR

BRNO 2012



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky a komunikačních technologií

Ústav telekomunikací

Bakalářská práce

bakalářský studijní obor

Teleinformatika

Student: Jakub Kolář **ID:** 120786

Ročník: 3 **Akademický rok:** 2011/2012

NÁZEV TÉMATU:

Bezpečnost operačních systémů pro mobilní zařízení

POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je podrobně rozebrat bezpečnostní rizika nejrozšířenějších operačních systémů pro mobilní zařízení (Android, iOS, Windows Phone 7, Symbian, RIM), možné hrozby, útoky a zneužití, které souvisí s touhle platformou. Výstupem práce bude vytvoření aplikace pro systém Android, která bude zasílat citlivá uživatelská data na vzdálený server. Popsány budou metody jakými lze aplikaci do zařízení nahrát bez vědomí uživatele. Výsledkem práce bude také návrh jak případné nedostatky vyřešit. Výstupem práce bude také laboratorní úloha zabývající se touto problematikou.

DOPORUČENÁ LITERATURA:

[1] DUNHAM, Ken. Mobile Malware Attacks and Defense. [s.l.] : Syngress Publishing, Inc, 2008. 440 s.ISBN 1597492981.
[2] MURPHY, Mark L. Android 2: průvodce programováním mobilních aplikací. Vyd. 1. Brno: Computer Press, 2011, 375 s. ISBN 978-80-251-3194-7 (BROž.).
[3] HOOG, Andrew. Android Forensics : Investigation, Analysis and Mobile Security for Google Android. [s.l.] : Syngress Publishing, Inc, 2011. 432 s. ISBN 1597496510.

Termín zadání: 6.2.2012 **Termín odevzdání:** 31.5.2012

Vedoucí projektu: Ing. Martin Rosenberg

prof. Ing. Kamil Vrba, CSc.

předseda oborové rady

UPOZORNĚNÍ:

ABSTRAKT

Tato bakalářská práce vysvětluje pojmy týkající se bezpečnosti mobilních zařízení. První kapitoly seznamuji s tím, jak se vyvinuli mobilní operační systémy a jaké typy se na dnešním trhu nacházejí. Jsou zde vysvětleny typy základních útoků a bezpečnostních trhlin. Dále jsou v práci rozebrány nejznámější a nejrozšířenější operační systémy, počínaje jejich základními popisy, architekturou a následně bezpečností. Následující kapitola obsahuje porovnání z hlediska bezpečnosti mezi jednotlivými operačními systémy. V osmé kapitole této práce je základní popis a vysvětlení funkcí samotného kódu vytvořené škodlivé aplikace. Kapitola obsahuje aplikací využité slabiny operačního systému Android, možnosti šíření škodlivých aplikací a také případnou ochranu před napadením. V poslední kapitole je uvedena laboratorní úloha, která seznamuje studenty s operačním systémem Android a zaměřuje se na poznání nevhodně použitých protokolů při komunikaci aplikací s internetem.

KLÍČOVÁ SLOVA

Mobilní operační systémy, zabezpečení, malware, Symbian OS, Android, iOS, Windows Phone 7, smsThief.

ABSTRACT

This term paper explains the concepts related to security of mobile devices. The first chapter acquainted with how to develop mobile operating systems and what types are found on the market today. Then the work explains the basic types of attacks and security vulnerabilities. Further work is discussed in the most famous and most popular operating systems, from their basic descriptions, architecture, and then security. In the eighth chapter of this work is a basic description and explanation of the function code itself created by malicious applications. The chapter contains weaknesses of the Android operating system used by the application, opportunities spread of malware and protect against possible attack. In the last chapter is contained laboratory task that acquaints students with the Android operating system and focuses on knowledge inappropriately protocols used to communicate with the Internet applications.

KEYWORDS

Mobile operating systems, security, malware, Symbian OS, Android, iOS, Windows Phone 7, smsThief.

KOLÁŘ, J. Bezpečnost operačních systémů pro mobilní zařízení. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2012. 56 s. Vedoucí bakalářské práce Ing. Martin Rosenberg.

PROHLÁŠENÍ

Prohlašuji, že svou semestrální práci na téma Bezpečnost operačních systémů pro mobilní zařízení jsem vypracoval samostatně pod vedením vedoucího semestrální práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené semestrální práce dále prohlašuji, že v souvislosti s vytvořením této semestrální práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne ....

(podpis autora)

PODĚKOVÁNÍ

Děkuji vedoucímu bakalářské práce Ing. Martinu Rosenbergovi za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé bakalářské práce.

V Brně dne ....

(podpis autora)

OBSAH

SEZNAM OBRÁZKŮ

ÚVOD

Mobilní zařízení jako taková jsou mezi lidmi poměrně krátkou dobu. Jejich historie nesahá tak hluboko do historie jako jiné produkty. Avšak za tuto krátkou dobu se byly schopné dostat mezi nutné věci mnoha lidí.

V dnešní době již existuje na trhu spousta typů mobilních zařízení počínaje klasickými či dotykovými mobily až po messengery a chytré telefony. Toto dělení je založené na konstrukci a funkcích telefonu. Klasické mobily jsou zařízení bez dotykové obrazovky s mobilní klávesnicí. Dotykové telefony zase disponují dotykovou obrazovkou, ale ve většině případů nejsou tak výkonné. Messengery jsou zařízení s plnou QWERTY klávesnicí, tak jak se používá u počítačů. Chytré telefony jsou již velice výkonná zařízení, u kterých není důležitá konstrukce zařízení (převážně však dotykové), a proto se pro jejich funkci používají tzv. *mobilní operační systémy*.

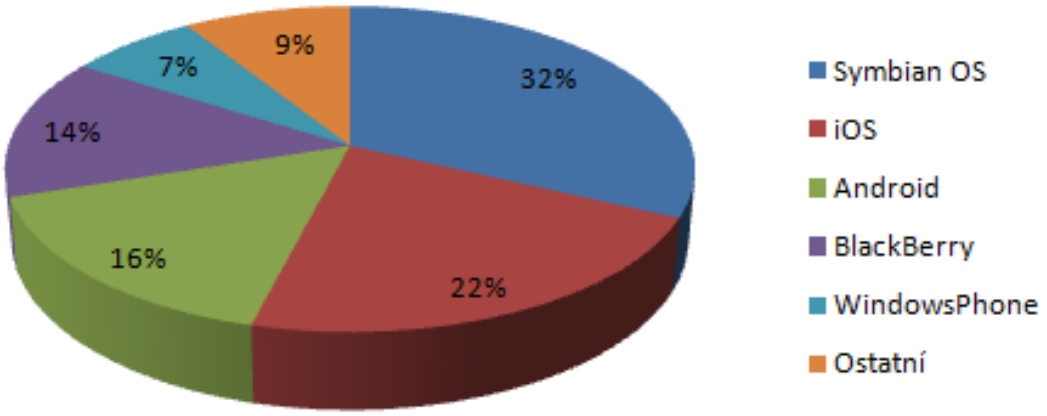
Mobilní operační systém pro chytré telefony se od systémů pro ostatní mobilní zařízení odlišuje svými funkcemi a možnostmi. Chytré telefony jsou díky nim schopné připojovat se k různým sítím (např. internet) stahovat si různé aplikace s různými funkcemi jako jsou různé hry, programy na promítání filmů, úpravu filmů. V podstatě již dnes téměř vše co je možné dělat na klasických počítačích.

Tyto rozsáhlé možnosti však činí mobilní zařízení zranitelné z hlediska zabezpečení. Je proto velice důležité, aby mobilní operační systémy byly velmi dobře zabezpečené, jinak by mohlo dojít k odcizení uživatelových osobních dat nebo by mohl být určitým způsobem okraden.

Tato práce se proto zaměřuje na zabezpečení nejpokročilejších a nejrozšířenějších operačních systémů. V prvních kapitolách práce jsou uvedeny slabiny a útoky již dnes známé pro mobilní operační systémy. Dále pak určitá prevence před dnešním škodlivým softwarem z pozice uživatele a následně popisy zabezpečení čtyř nejčastějších mobilních operačních systémů v České republice. Následně je uvedené porovnání bezpečnosti mezi zmíněnými mobilními operačními systémy. V posledních kapitolách práce je popsána vytvořená škodlivá aplikace zabývající se znázorněním určitých slabin v bezpečnosti mobilního operačního systému Android. Jsou zde vysvětleny využité slabiny, možnosti šíření podobných škodlivých aplikací a možnosti ochrany vlastního zařízení. Na závěr, práce obsahuje laboratorní úlohu, jejíž funkcí je seznámení studentů s operačním systémem Android a poukázání na nedostatečnou bezpečnost při komunikaci mezi aplikacemi a internetem.

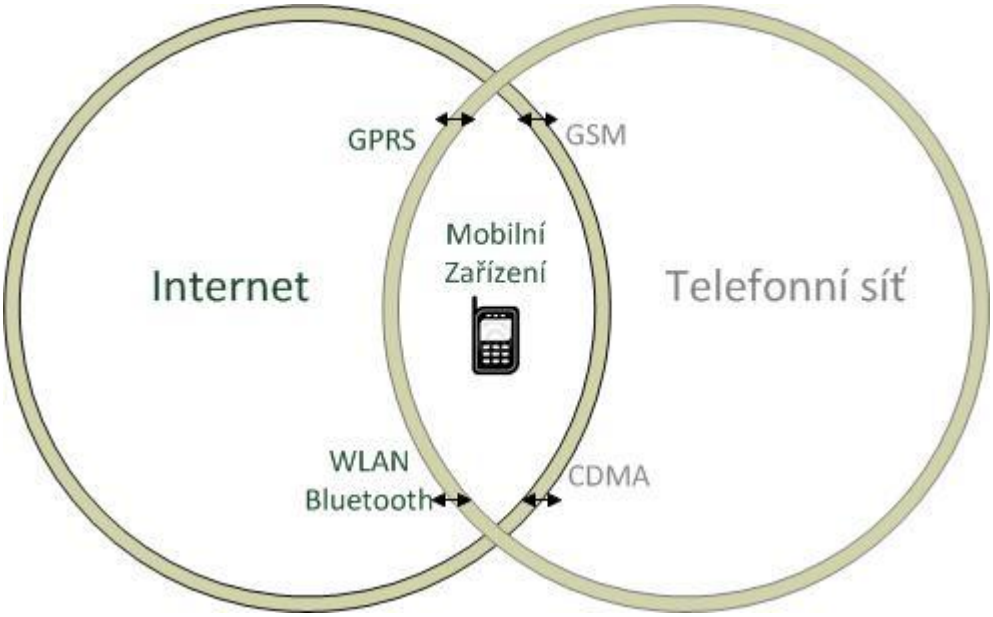
1. MOBILNÍ ZAŘÍZENÍ A JEJICH SLABINY

Mobilní telefony, zařízení vymyšlená za účelem spojit lidi na velké vzdálenosti bez nutnosti nacházení se na určitém místě. Tyto zařízení se již dnes vejdou každému do kapsy, a proto jsou lidé přístupní téměř kdekoliv a kdykoliv. Mobilní telefony v dnešní době jsou již nedílnou součástí lidské společnosti a nikdo si již nedokáže představit život bez jejich použití. I přesto, že jejich historie nesahá tak moc daleko do té lidské, dokázali se neuvěřitelně vyvinout a posunout kupředu. Mobilní telefony, tak jak jsou dnes známy, už dávno neslouží pouze pro telefonování a využívání služby SMS (Short Message Service). Dnešní mobilní zařízení se již nachází na úrovni sofistikovaných počítačů, jejichž výpočetní výkon je téměř srovnatelný a díky tomu jsou schopny provozovat nejrůznější aplikace využívající nejrůznějších služeb. Tyto mobilní zařízení jsou dnes označovány pod pojmem *smart*-phone, neboli chytrý telefon. Chytré telefony využívají nejrůznější mobilní operační systémy (OS). Mobilních operačních systémů je v dnešní dobře celá škála, za zmínku stojí pouze ty nejrozšířenější a nejznámější jako jsou Symbian OS, iOS, Android, BlackBerry OS a WindowsPhone7. Celkové zastoupení těchto jednotlivých mobilních operačních systémů na celosvětovém trhu, je znázorněno v následujícím grafu (viz Obr. 1.1). Tento graf však zahrnuje všechna zařízení fungujících na těchto operačních systémech včetně tabletů, klasických telefonů, chytrých telefonů, multimediálních zařízeních apod.



Obr. 1.1: Procentuální podíl mobilních operačních systémů na celosvětovém trhu [1].

U těchto chytrých telefonů jsou možnosti už tak rozsáhlé, že by se velmi těžko vypisovaly všechny jejich funkce, které jsou schopné provést. Za zmínku však stojí jedna z nejdůležitějších, kterou je schopnost připojení se k internetu. To že jsou dnešní mobilní zařízení schopná připojit se k internetu, mísí dvě různá prostředí do jednoho (viz Obr. 1.2). S internetem nepřichází pouze výhody, ale také různá úskalí co se uživatelských mobilních dat týče. Avšak internet není zdrojem všech možných útoků, většina je s ním právě spojena.



Obr. 1.2: Znázornění propojení mobilní sítě s internetem [2].

1.1 Typy útoků

Na mobilní zařízení již v dnešní době existuje celá řada různých útoků. Tyto útoky se liší ve způsobu provedení ale i v následných škodách. Jedná se o útoky, které můžou postihnout většinu telefonů, ne jenom dnešní chytré telefony, protože se především jedná o využívání chyb v různých komunikačních rozhraních jako je například *Bluetooth* nebo *Wifi*. V této kapitole jsou uvedeny známé útoky z minulosti s jejich základními popisy [2].

Hacking Defaults

Technika používaná k proniknutí do zařízení nebo softwaru, který využívá znalosti o standardních heslech, nastaveních a konfiguracích.

Denial-of-Service (DoS)

Útok navrhnutý k narušení nebo odepření použití zařízení, služby nebo sítě. Jsou známé i typy těchto útoků kdy po narušení došlo k následnému výpadku nebo odepření použití.

Exploit

Software nebo sekvence příkazů využívající chyby, závady či nějaké slabiny v systému s cílem způsobit nežádoucí nebo neočekávané chování systému. Exploit však může být také neúmyslný, například když uživatel provede nějakou akci, při které naruší kód právě probíhajícího neodladěného systému. Tyto chyby se opravují vydáváním různých aktualizací s opravou daného kódu pro docílení odladění systému.

Blover/II

Aplikace vytvořená v programovacím jazyku java určená pro odcizování dat. Pomocí této aplikace můžeme stáhnout telefonní kontakty, přijaté SMS apod. Bohudík může být použita pouze na zařízeních pracujících s J2ME (Java 2 Platform, Micro Edition), což je java edice pro bezdrátová a mobilní zařízení. Aplikace využívá pro stažení dat rozhraní bluetooth.

Bluebug

Tento útok využívá slabin v ochraně bluetooth. Pomocí něho je útočník schopný generovat odchozí hovory, posílat SMS, využívat mobilního připojení k internetu ale také zamezit uživateli v používání všech možných typů komunikací na jeho zařízení.

BlueBump

I tento útok využívá slabin v ochraně bluetooth. Útočník využívá určitých postupů k získání připojení na napadené zařízení, které si díky chybě v ochraně smaže svůj ověřovací klíč, ale útočníka nechá stále připojeného.

BlueChop

Jeden z DoS útoků vyvinutý k narušení piconet sítě. Útočník zde využívá zařízení nepřipojeného do piconet sítě (tzv. otroka) ke kontaktování zařízení, které tuto síť řídí (tzv. mistra). Otrok si zde však myslí, že má být součástí sítě a tím zmate mistrův vnitřní stav, což vede k narušení piconet sítě. Předpokladem k tomuto útoku je mistrova schopnost připojení více zařízení najednou. Pro úplnost: piconet síť je připojení více zařízení bez předběžného návrhu sítě pomocí bluetooth.

BlueDump

Technika používaní k narušení ověřovacího klíče mezi dvěma zařízeními pomocí bluetooth. V tomto případě útočník zfalšuje adresu jednoho z účastníků a pokusí se připojit k jinému, který zažádá o autorizaci. Nyní odešle útočník negativní odpověď, což v některých případech způsobí, že si autorizující zařízení smaže svůj vlastní klíč a vstoupí do párovacího módu.

Bluejacking

Útok podobný zasílání nežádoucí pošty (tzv. spamu). Nežádoucí zpráva je zasílána ostatním zařízením pomocí rozhraní bluetooth, což síť zahlcuje a narušuje tak párování mezi těmito zařízeními.

Blueprinting

Je metoda pomocí, které dokážeme zjistit detaily o zařízeních se zapnutým rozhraním bluetooth v našem okolí. Detaily jako model, výrobce a také zda zařízení má bezpečnostní trhliny ve svém přenosovém rozhraní bluetooth.

Bluesmack

Další z bluetooth útoků kde se zasílá obrovský ping paket, aby se docílilo DoS u cílového zařízení. Velmi podobný útoku „Ping of Death“ používaného u předchozí verze počítačového operačního systému Windows 95.

Bluesnarf/++

Bluesnarf nejspíše nejznámější bluetooth útok vzhledem k tomu, že využívá jedné v největších slabin tohoto rozhraní. Útočník se musí připojit pomocí OBEX Push Profile (OPP), který byl určený pro jednoduchou výměnu vizitek, kontaktů a jiných objektů. Pomocí Bluesnarf poté zruší párování a připojí se na neověřený kanál, kde získá čtecí a zapisovací přístup k napadenému zařízení.

Car Whisperer

Tento útok zneužívá osobního identifikačního čísla (tzv. PIN) k připojení k autům. Umožňuje tak útočníkovy například nahrávat nebo stahovat audio záznamy.

HelloMoto

Tento útok je kombinací Bluesnarf a BlueBug útoků. Po spárování útočník odešle vCard a akci přeruší, tím získá ověření od napadeného zařízení a poté již pomocí AT příkazů získá kontrolu nad napadeným zařízením.

Vzhledem k tomu, že dnešní zařízení již používají složité operační systémy, vznikl zde nový druh útoků. Tyto útoky jsou tzv. škodlivý software, který je většinou zaměřován na konkrétní operační systém a jeho chyby. Tento škodlivý software se nazývá Malware [3].

1.2 Mobilní Malware

Malware zkratka z anglického malicious software znamená v překladu škodlivý program. První mobilní malware se u mobilních zařízení začal objevovat v roce 2000 a od této doby stále roste. Avšak největší nárůst byl zaznamenán v roce 2004, kdy byl vytvořen první mobilní červ jménem *Cabir*. Tento škodlivý program se velmi rozšířil a nejvíce zasáhl platformu Symbian. Od té doby se však vyvinulo několik nových mobilních operačních systémů, u kterých se vývojáři zaměřili více na bezpečnost. Přesto i na tyto novější mobilní operační systémy jako například iOS pro iPhone najdeme dnes spoustu různých útoků. Obecné výrazy a jejich popisy jsou uvedeny v následující kapitole 1.2.1 [2].

I přesto, že mobilní i počítačový malware jsou vytvářeny za účelem škodit, krást informace a znemožňovat užívání zařízení, liší se v mnoha aspektech týkajících se slabin a naopak silných stránek. Mezi tyto aspekty se řadí následující:

- Mobilní zařízení jsou téměř stále připojené k síti, kde se neustále mění sousední zařízení v závislosti na přesouvání.

- Počítače se ve většině případů nacházejí v síti, kde se sousední zařízení mění jen zřídka.

- Slabiny objevené u počítačů mohou být opraveny vydáním aktualizace, která se uživateli nabídne po připojení k internetu.

- U mobilních zařízení se nepočítá s tím, že se každý uživatel připojuje se zařízením k internetu a proto je velice náročné vydávat aktualizace, které by slabiny opravovaly.

Vzhledem k tomu, že s pokročilými technologiemi se tyto rozdíly vytrácejí, pro výrobce je stále důležité brát je v potaz, aby docílili vysoké bezpečnosti u mobilních zařízení.

1.2.1 Výrazy mobilních malware útoků

Ad / Spyware

Jsou to programy, které jsou nainstalovány na uživatelovo zařízení bez jeho vědění a většinou je velice těžké je odhalit. Spyware programy monitorují uživatelovo zařízení, ale také sbírají různé druhy informací, například jaké webové stránky uživatel navštěvuje apod. Toho pak zneužívají ve svůj prospěch, jsou schopné zahlcovat zařízení, zpomalovat rychlost připojení k internetu a posílat získané data na vzdálený server.

Payload

Payload je jedna ze základních akcí škodlivého softwaru. Označují se jí data škodlivého softwaru, která jsou úspěšně dopravena na napadené zařízení.

Rogue software

Tento útok může být přeložen jako zlodějský nebo nepoctivý software. Jedná se o program, který se svým chováním snaží přesvědčit uživatele o tom, aby si zakoupil nějaký jiný program. Toto chování se projevuje chybnými výsledky při kontrole zařízení nebo různými upozorněními o tom, že zařízení není v pořádku.

Trojan

Software pojmenovaný po bájném trojském koni, jehož využitím je inspirován. Tento software se tváří jako něco čím není, tudíž uživatel si ho ve většině případů nevědomky uloží na zařízení sám, kde již pak škodí dle úmyslů útočníka. Trojan se sám o sobě nekopíruje.

Virus

Škodlivý software, který infikuje data v napadeném zařízení za účelem se co nejvíce rozšířit.

Worm

Neboli červ v překladu. Červ v napadeném zařízení kopíruje sám sebe za účelem se co nejvíce rozšířit. Nejznámějším červem v historii mobilních zařízení se stal zmiňovaný *Cabir*, který využíval bezpečnostní slabiny v rozhraní bluetooth. Tento červ se po jeho stažení na zařízení zkopíroval mezi systémové soubory, poté zapnul rozhraní bluetooth a začal se odesílat na všechny zařízení v okolí. I přesto, že uživatel zařízení musel přijmutí červa potvrdit, nebylo toho těžké dosáhnout, protože se *Cabir* tvářil jako bezpečnostní aktualizace.

2. OBECNÁ PREVENCE PROTI MODERNÍM ÚTOKŮM

Určitě v každém mobilním operačním systému je možné najít několik bezpečnostních rizik. Šikovný hacker, který se bude chtít dostat do zařízení za účelem škody nebo krádeže si vždy najde cestu. Obyčejní lidé však nesmějí být paranoidní a předpokládat, že někdo takhle zkušený a zdatný se nebude snažit okrást zrovna je. Pokud se chce kdokoliv stát spokojeným uživatelem *smart-phonu,* měl by znát možná rizika a zajistit si určitou prevenci.

Za základní prevenci proti napadení zařízení se považuje informovanost uživatele o možnostech a úskalích o daném operačním systému. Zde se jedná především o stahování aplikací, volbě vhodného antivirového programu, volbě vhodného internetového prohlížeče, nebo také aktualizace samotného operačního systému.

2.1 Stahování aplikací

Jako jednu z prvních věci při koupi *smart-phonu* se téměř každý uživatel připojí na internet a začne stahovat různé aplikace, aby obohatil svoje zařízení. V dnešní době téměř každý mobilní operační systém má nějaký druh svého internetového obchodu, ze kterého mohou jeho uživatelé stahovat různé aplikace. Bohužel ne všichni výrobci se mohou chlubit bezpečností všech aplikací, které jsou v jejich obchodě k dispozici. Dále je také možné stahovat aplikace z neznámých a neověřených zdrojů, z různých internetových stránek apod. Uživatel by si měl být jistý věrohodností aplikace, jinak je velmi pravděpodobné, že si škodlivou aplikaci do svého mobilního zařízení stáhne. Nejbezpečnějším řešením je stahovat aplikace pouze z oficiálního obchodu daného operačního systému a u každé aplikace si přečíst názor ostatních uživatelů o dané aplikaci.

2.2 Antivirus

V poslední době lidé začínají používat svoje mobilní zařízení jako jejich klasické počítače. Představa, že by si smazali antivirový program a dále používali notebook, je velice riskantní, protože každý ví, že je velmi mnoho škodlivého softwaru na téměř všechny druhy počítačových operačních systémů. Tyto hrozby však už postihují také mobilní zařízení. Spousta dnešních uživatelů používá jejich *smart-phone* jak na práci, tak také na zábavu. Zde si potom nemohou být jisti, že stránka, kterou navštěvují nebo aplikace, kterou si nainstalují, neobsahuje škodlivý software. Proto je vhodnou volbou mít nainstalovaný antivirový program pro případ, že by nastala krizová situace. Při instalaci antivirového programu je však také velmi důležité informovat se o funkcích daného antiviru, protože jsou zde i antivirové programy, které pouze sledují provoz na zařízení, ale s problémem si nijak neporadí.

2.3 Internetový prohlížeč

Správná volba internetového prohlížeče pro klasického uživatele není tak důležitá, vzhledem k tomu, že mobilní webové prohlížeče, dostupné na různé zařízení, si jsou velmi podobné. I přesto je důležité si uvědomit, že jejich zabezpečení není tak propracované, jak je tomu u jejich „větších“ příbuzných u stolních počítačů. Mobilní webové prohlížeče jsou poněkud složitější než většina ostatních aplikací. Mají přístup k celé řadě ovládacích prvků na mobilním zařízení a tím jsou způsobeny bezpečnostní slabiny, pomocí kterých se stávají nejlehčím způsobem, jak dálkově ovládnout chytrý telefon. Pro uživatele používajícího chytrý telefon pro práci je už tohle riziko větší a měl by se informovat a případně zabezpečit. Pro firmy používající více jak padesát mobilních zařízení pro práci se doporučuje kontaktovat dodavatele mobilního software za účelem poskytnutí takové aplikace, která bude všechen obsah udržovat zabezpečený a v aktuálním stavu.

2.4 Aktualizace

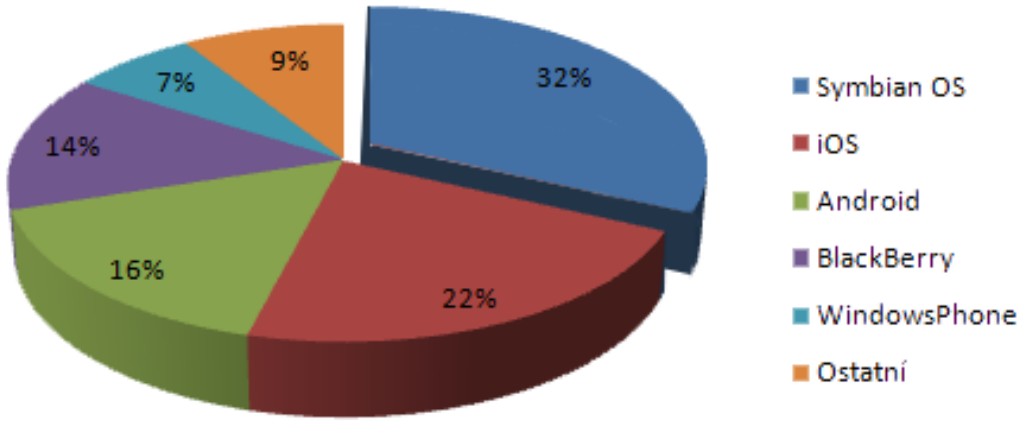
Aktualizace operačního systému může být u dnešních chytrých telefonů v mnoha případech ošemetnou záležitostí. Nejedná se zde o oficiální aktualizaci poskytovanou výrobcem, ale především o různé možnosti poskytnuty domácími kutily. Naskytují se zde možnosti, jako je nahrání jiného operačního systému, nainstalování vyšší verze, která však není podporovaná apod. To uživatele zláká a oni tuto volbu zkusí. To však s sebou nese několik podstatných nebezpeční a nevýhod. Například při přehrání operačního systému uživatel téměř okamžitě ztrácí záruku na svoje zařízení, tedy za předpokladu, že si původní oficiální systém nenahraje zpět. Další věcí je, že tyto systémy nejsou optimalizovány. Což má za následky obrovské nevýhody jako jsou rychlejší vybíjení baterie, nefunkčnost některých prostředků (nejčastěji rozhraní wifi, bluetooth), chybné probouzení telefonu, nefunkčnost některých tlačítek apod. Jako nebezpečí je zde pak samotná instalace, která při nějakém chybném kroku může způsobit nevratnou nefunkčnost celého zařízení. Tyto kroky se proto důrazně nedoporučují, protože se zde může uživatel stát sám sobě škůdcem.

3. OPERAČNÍ SYSTÉM SYMBIAN

3.1 Historie Symbian OS

Mobilní operační systém Symbian je jeden z nejstarších a momentálně nejrozšířenější operační systém na celosvětovém trhu. První zmínky o tomto systému padli v roce 1998 kdy se společnosti Nokia, Ericsson a Psion dohodli na jeho vývoji, také Motorola projevila zájem o tento systém, což ještě vývoj posílilo. Jeho základem byl 16ti bitový organizér *EPOC*, který dobře zvládal multitasking (současný běh několika aplikací), vyráběný jednou z partnerských společností Psion. První telefon byl představen v roce 2000 a díky obrovské výhodě, kterou byla možnost vývoje aplikací pro danou platformu výrobci třetích stran, se během následujících let rozšířil mezi mnoha výrobci [6].

S probíhajícími léty se systém stále vyvíjel a vycházeli nové verze, což vedlo k výrazným změnám. Velkou revolucí bylo představení Symbianu v^2, jehož změny již byli tak rozsáhlé, že zde nebyla možná zpětná kompatibilita. Tedy aplikace postavené pro starší verze nefungovaly pro zmiňovanou verzi v^2. Systém se však stal částečně *open-source,* neboli kód operačního systému se stal přístupný pro všechny, jež byli součástí Symbian Ltd., jehož většinovým vlastníkem již byla společnost Nokia (99,95% akcií).



Obr. 3.1: Podíl Symbian OS na celosvětovém trhu mobilních operačních systémů [1].

V roce 2010 byla představena verze Symbian^3. S touto verzí systému se již systém stal plně otevřeným, tedy že kód je přístupný všem. Na tuto poslední verzi operačního systému Symbian již vyšli dvě aktualizace označované *Anna* a *Belle*, které systém dále rozšiřují o moderní technologie jako je podpora rozhraní *NFC* (Near-Field-Commucitaion). Jak to však bude s touto platformou dále, není jasné, vzhledem k tomu, že vlastník a momentálně jediný výrobce mobilních zařízení Nokia se začíná zaměřovat na zařízení s *Windows Phone 7*. I přesto je stále Symbian nejrozšířenějším mobilním operačním systémem na celosvětovém trhu (viz Obr. 3.1) se 32% zařízení z celku [7].

3.2 Architektura Symbian OS

Symbian je mobilní operační systém určený výhradně pro mobilní zařízení. Běží na jádře EKA2 (EPOC Kernel Architecture 2), které pracuje v reálném čase. Základní jádro zpracovává jak uživatelské aplikace, tak funkce telefonu. Pro dosažení větší funkčnosti, maximální odolnosti a odezvy obsahuje mikrojádrovou architekturu, která se stará o ovladače pro dané zařízení, telefonní síť a správu systémové podpory.

Model operačního systému Symbian poskytuje multitasking, multithreading (umožňuje existenci více funkcí v rámci jednoho procesu) a ochranu paměti. Pro jejich splnění a dosažení maximální spolehlivosti je rozdělený na pět vrstev (viz Obr. 3.2) [8].

Obr. 3.2: Model vrstev operačního systému Symbian [8].

Vrstva uživatelského rozhraní

V Symbian OS je uživatelské rozhraní odděleno od zbytku operačního systému, což umožňuje výrobcům vytvořit si vlastní prostředí telefonu. Například u verze Symbian v^2 je vytvořeno grafické prostředí *AVKON* známé jako S60 (série 60 uživatelské prostředí), které bylo použito u mobilních telefonů Nokia. Dnes se tato společnost snaží prosadit rozhraní *Qt Quick*, které již umožňuje vytvoření vizuálně bohaté obrazovky pro plně dotyková zařízení.

Vrstva aplikačních služeb

Tato vrstva poskytuje nezávislou podporu pro rozhraní jednotlivých aplikací, tedy se stará o možnosti různého zobrazení v různých aplikacích. Obsahuje personálního informačního manažera (PIM), základy pro zprávu kancelářských aplikací, ale také multimediální protokoly pro funkčnost těchto aplikací.

Vrstva služeb operačního systému

Kromě obecných služeb operačního systému tato vrstva ještě obsahuje komunikační, multimediální a připojovací služby.

Vrstva základních služeb

Tato vrstva obsahuje knihovny nižších vrstev, které umožňují abstrakce jádra hardwaru.

Vrstva služeb jádra

Tato nejnižší vrstva obsahuje jádro samotného operačního systému spolu s fyzikálními a logickými ovladači pro daný hardware.

3.3 Zabezpečení Symbian OS

Zabezpečení, určitě jedna z nejvíce důležitých vlastností každého operačního systému. Avšak nejen u Symbian OS nedokonalá záležitost, i přesto jak moc o ni výrobci usilují. U tohoto systému je jedním z důvodů jeho otevřenost. S otevřeným systémem může totiž ke zdrojovému kódu přistoupit každý včetně záškodníků. Se znalostí zdrojového kódu se mnohem snáze pro zmiňované záškodníky tvoří škodlivý software. Další nevýhodou je rozšířenost tohoto operačního systému, která tak láká více záškodníků, protože jejich škodlivý software se lépe rozšíří, což je povětšinou účelem této skupiny lidí. Vývojáři se však snaží řídit následujícími kroky k tomu, aby dosáhli dostatečného zabezpečení a uživatelé tak byli s jejich zařízeními spokojeni [7].

3.3.1 Požadované vlastnosti zabezpečení

Soukromí

Soukromí u mobilního operačního systémů Symbian je vlastnost, která při manipulaci s uživatelskými osobními daty zajišťuje, že nedojde k jejich odhalení. Různé druhy informací mohou být považovány za soukromé. Ať už se jedná o adresář kontaktů, záznamy v kalendáři nebo komunikaci při hovoru. I přesto, že jednotlivé vlastnosti jsou neméně důležité pro dosažení vynikající bezpečnosti, je tato vlastnost považována za základní kámen bezpečnostní architektury Symbian OS.

Spolehlivost

Je schopnost systému zajistit funkčnost zařízení po celou dobu životnosti. Je velice úzce spjatá s dostupností, protože pokud zařízení vykazuje chyby při navazování spojení, uživatel se tak stane nedostupný. Zde se však nejedná pouze o správu hovorů, jedná se o všechny aspekty funkčnosti zařízení. Pro dosažení spolehlivosti je důležité se při návrhu bezpečnostní architektury zaměřit na ochranu kriticky důležitých systémů a konfiguraci zařízení. Je velice důležité zajistit, aby nebylo možné tyto části napadnout nebo upravit neoprávněnou osobou.

Obrana

K zajištění zachování funkčnosti všech zmiňovaných vlastností je velice důležité, aby byl systém schopný odolávat různým druhům útoků. Jedná se zde o ochranu proti mobilnímu *malwaru*. To je důvod začlenění této vlastnosti do bezpečnostní architektury.

Jednoduchost

I přes důležitost všech zmiňovaných bezpečnostních vlastností je prioritou zajistit, aby zůstali co nejméně viditelné. Většina uživatelů dnešních mobilních zařízení preferují, že nejsou otravováni různými rozhodnutími ohledně bezpečnosti, jako je tomu například u stolních počítačů. Je tedy důležité, aby systémy starající se o bezpečnost byly přizpůsobeny těmto požadavkům.

Důvěryhodnost

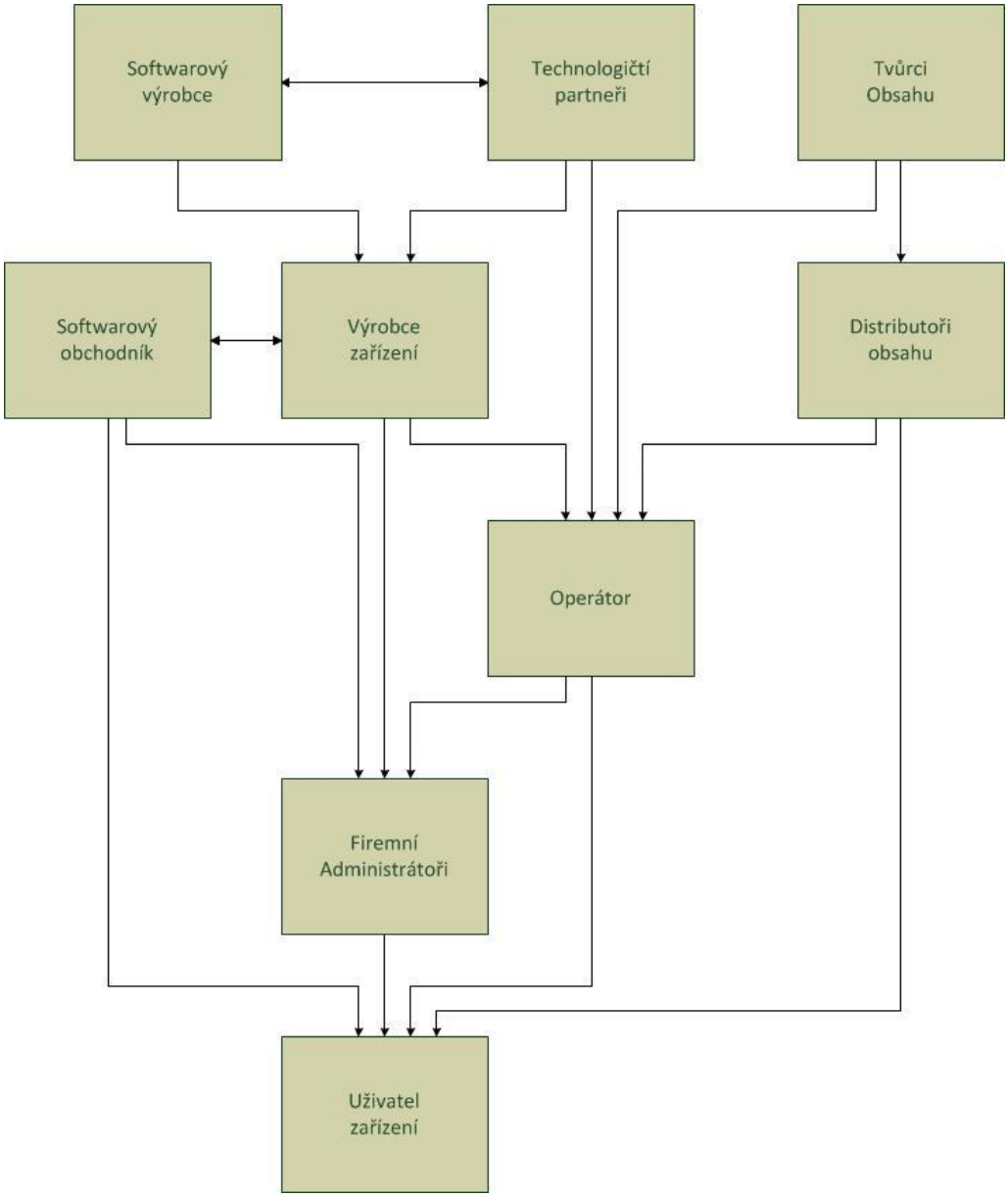
Velice ošemetná vlastnost u mobilních zařízení, přesto jí vývojáři Symbian OS uplatňují ve svém systému. Jedná se zde o výměnu důvěryhodných informací (např. telefonní číslo, bankovní zpráva apod.) mezi zařízením a ostatními sítěmi. Vývojáři se zde potýkali s filozofickou diskuzí ohledně důvěry. Nakonec bylo rozhodnuto, že mezi důvěryhodná uskupení, vzhledem k uživateli, bude považován výrobce zařízení a mobilní operátor. Ale vzhledem k pokroku ve vývoji za poslední desetiletí se zde začali vyskytovat různá úskalí. Typickým příkladem jsou rozšířené možnosti konektivity dnešních zařízení. Dnes si všichni uživatelé mobilních telefonů se Symbian OS mohou přijít do nějaké kavárny či jiného místa, kde se mohou připojit na veřejnou wi-fi síť. To však umožňuje možnost napadení za účelem získání důvěryhodných informací. Pro tyto situace byly v Symbian OS vytvořený protokoly, které umožňují zařízení chránit informace nezávisle na připojené síti.

3.3.2 Bezpečnostní architektura Symbian OS

Pro dosažení vysokých cílů bezpečnosti se vývojáři Symbian OS zaměřují na řešení dopravování všech možných dat (aplikace, hovory, kontakty apod.) od poskytovatelů přímo do rukou uživatele mobilního zařízení. Uvažuje se zde několik různých organizací včetně Symbian OS, které se zabývají poskytováním produktů pro mobilní telefony a skládá se z nich „řetězec hodnot“(viz

Obr. 3.3), který slouží pro vybudování bezpečné architektury. Je nutné si zde uvědomit, že mobilní zařízení se Symbian OS je koncovým bodem v síti a není možné, aby všechna bezpečnost byla právě na něm. Symbian OS zde staví základní kámen pro zmiňovaný řetězec hodnot jakožto softwarový výrobce a snaží se docílit perfektní architektury za účasti všech článků řetězce [7].

Pro maximální bezpečnost je nutné, aby hardware a software zařízení byl úzce svázán. Tedy aby zde nedocházelo k chybám nedorozumění kvůli špatné optimalizaci. Softwarový výrobce, technologičtí partneři a výrobce zařízení zde musí spolupracovat, aby bylo dosaženo nastavení všech bezpečnostních prvků, jako je například firewall zařízení, ale také aby se v průběhu používání neobjevili chyby, které by mohli být při testování přehlédnuty (stalo se u konkurenčního OS). Softwarový obchodníci, jimiž jsou zde myšleni obchodníci třetích stran, kteří vyrábějí různé aplikace pro daný OS, a tvůrci obsahu s jejich distributory (např. vydavatelé e-knih), se přímo nepodílejí na bezpečnosti zařízení tak, jak se dostane uživateli do ruky při koupi, ale i přesto jsou nedílnou součástí celkové architektury. Musí zaručit, aby od nich uživatelem koupené aplikace byly z bezpečnostního hlediska bezchybné, jinak by uživatel ztratil důvěru ke kupování těchto aplikací a za předpokladu uchránění se, by je přestal nakupovat. To by vedlo k úpadku OS, protože by uživatelé nemohli využívat jeho plný potenciál.



Obr. 3.3: Řetězec hodnot pro mobilní zařízení se Symbian OS [7]

Předposlední skupinou v řetězci jsou firemní administrátoři a operátoři. Ti zajišťují jak bezpečnost dat přenášených po síti, tak i správu životního cyklu aplikací, do které například patří zajišťování aktuálnosti, záplatování objevených chyb apod. Posledním článkem je pak samotný uživatel, jehož činy k dosažení bezpečnosti byly popsány v kapitole 2.

3.3.3 Odhalování bezpečnostních incidentů a následné reakce

Vývojáři Symbian OS považují jejich bezpečnostní architekturu, tak jak je navržená, za velice efektivní. I přesto zde jednou začas dochází k různým bezpečnostním obligacím či průrazům. Proto se zde musí počítat s tím, že každý incident, musí být eliminován.

Ve většině případů je první osobou, která incident zaznamená samotný uživatel zařízení. Předpokládá se, že uživatel po zjištění problému kontaktuje svého mobilního operátora nebo prodejce zařízení. Od tohoto bodu vše směřuje co nejrychleji na systémový server *US-CERT* (United States Computer Emergency Readiness Team), kde se analyzují všechny hlášené problémy. Zde se odhaluje původ daného problému a jeho následné vyřešení. Tento systém již není tolik vytížen v dnešní době, protože díky možnosti analyzování a následné nápravy chyb se vývojáři naučili jak některým chybám předcházet. To vedlo k vylepšení Symbian OS na úroveň ve které je dnes.

3.3.4 Příklady bezpečnostních chyb systému Symbian OS

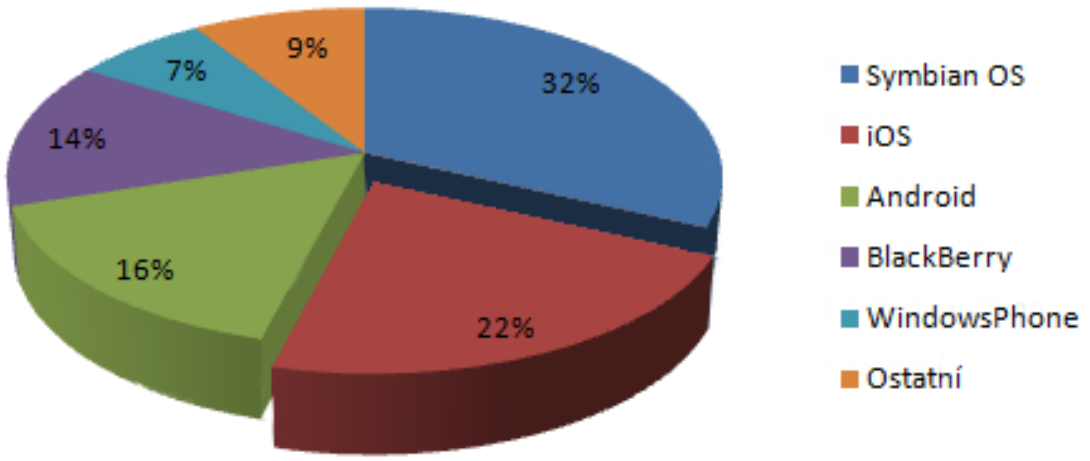
Jak je patrné z předchozích kapitol, Symbian OS nepatří mezi nejlépe zabezpečené mobilní operační systémy. Toto tvrzení sice nemůže platit pro všechny verze, protože nový Symbian Anna a Belle jsou již lépe zabezpečené a nachází se na ně významně méně škodlivého softwaru než na předchozí verze. To však stále umožňuje obrovský výběr mezi škodlivým softwarem.

Mezi typický škodlivý software patřící k zástupu mobilního malwaru na starší verze Symbian OS patří například tzv. *SymbOS/Hobbes.A*. Tento trojský kůň po instalaci na zařízení zobrazí dialog navádějící uživatele k potvrzení instalace Symantec antiviru pomocí restartování zařízení. Samozřejmě trojský kůň neobsahuje žádný antivirový program a po restartování zařízení dojde k vyřazení aplikačního menu, které slouží ke správě aplikací.

Dalším úspěšným příkladem škodlivého útoku na platformu Symbian je Botnet virus. Virus se nabourává do systému zařízení a odesílá všem uživatelovým kontaktům SMS zprávy, které obsahují odkazy na webové stránky obsahující další škodlivý software včetně daného viru. Aby nedošlo k odhalení tak se virus prezentuje jako herní aplikace a všechny odeslané SMS po sobě vymazává.

4. OPERAČNÍ SYSTÉM IOS

Operační systém iOS je mobilní operační systém vytvořený společností Apple. Co se týče mobilních zařízení, je jeden z nejrozšířenějších a nejpopulárnějších i přesto, že je vyráběn pouze jeden typ chytrého telefonu, který tento systém obsahuje. Kromě chytrých telefonů označovaných *iPhone*, je systém používán v multimediálních přehrávačích a tabletech (iPod, iPad), kde si však zachovává stejnou podobu. IOS je uzavřený systém, avšak na světě je velmi silná hackerská komunita, která se snaží tento systém různě vylepšit a zpřístupnit. Co se mobilních zařízení týče, iOS se drží na druhé pozici nejrozšířenějších mobilních operačních systémů s 22 % zastoupením s celkového počtu (viz Obr. 4.1).

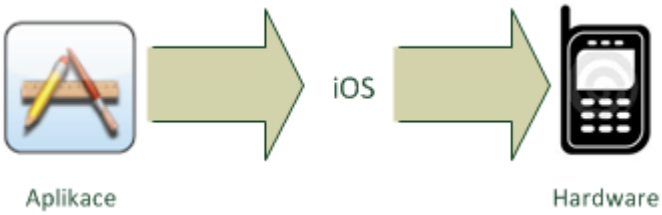


Obr. 4.1: Podíl iOS na celosvětovém trhu mobilních operačních systémů [1].

Mobilní operační systém iOS byl představen 9. ledna 2007 ve verzi 1.0.0. První zařízení obsahující tento software byl iPhone první generace (dnes označovaný 2G). Systém se od té doby vylepšil až do verze iOS 5.0.1, přičemž každý z updatů přinesl pro systém nové možnosti a opravy v zabezpečení [9].

4.1 Architektura systému iOS

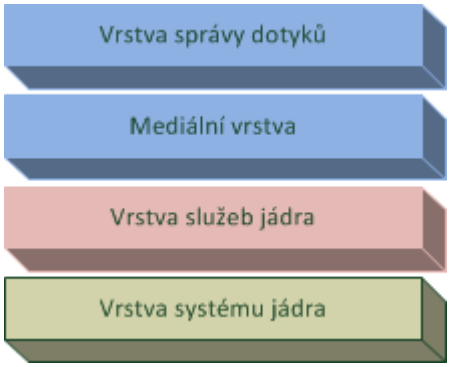
Architektura operačního systému iOS odvozená ze základu *Mac OS X*, má vzhledem k uzavřenosti operačního systému k dispozici popsány pouze nejvyšší stupně této architektury pro developerské účely. Na nich systém iOS funguje jako zprostředkovatel mezi hardwarem zařízení a aplikacemi, tedy softwarem (viz Obr. 4.2). Aplikace, které jsou tedy vytvořeny a nahrány do zařízení, ať už od samotného výrobce nebo aplikace třetích stran, nemají přímý přístup k hardwaru zařízení. Místo toho aplikace komunikují s hardwarem pomocí sady dobře definovaných systémových rozhraní, které chrání aplikace před chyby způsobené náhlými změny (hovor, SMS apod.).



Obr. 4.2: Ukázka iOS jako zprostředkovatelem mezi aplikacemi a hardwarem

Implementace systémových technologií u iOS je rozdělena do sady vrstev (viz Obr. 4.3). Spodní vrstvy obsahují základní služby a technologie, na které všechny aplikace spoléhají. Vyšší vrstvy obsahují více sofistikované služby a technologie, které mají napomáhat developerům ve vytváření aplikací. Tyto služby činí vytváření aplikací jednodušší tím, že redukují množství kódu, které je nutné psát. Ačkoliv vyšší vrstvy umožňují využívání technologií nižších vrstev, nezamezují přímou práci s nimi. A je zde možné pracovat přímo s nižšími vrstvami a využívat některých aspektů, které vyšší vrstvy nezahrnují.

Vrstva správy dotyků obsahuje základní rámce pro vytváření aplikací u iOS. Mezi takové patří například multitasking, ochrana dat, tisk na vzdálené zařízení, rozpoznávání gest apod. Nižší vrstva označovaná jako mediální vrstva obsahuje vše potřebné pro poskytování plnohodnotné multimediální zkušenosti, tedy grafické, audio a video technologie. Nejnižší vrstvy už obsahují základní systémové služby, které všechny aplikace využívají. Mezi ně patří například synchronizace se službou iCloud nebo nakupování přes internetový obchod Apple Store (služba pro nákup a stahování aplikací) [10].



Obr. 4.3: Sada vrstev poskytovaných technologií u systému iOS [10].

4.2 Zabezpečení systému iOS

Zabezpečení je jednou z velice silných stránek systému iOS. Je tomu proto, že se jedná o uzavřený systém a uživatel zde nemá takové možnosti interakce s nastavením mobilního zařízení, jako je tomu u konkurence. Na iPhone nebo jiných zařízeních podporujících systém iOS není možné instalovat aplikace jinak než přes oficiální internetový obchod Apple Store. Samozřejmě toto nastavení se již dnes dá odstranit pomocí tzv. *jailbreak*, ale není doporučováno pro klasické uživatele, protože

s jeho instalací se odstraňují určité prvky zabezpečení. V neupraveném systému iOS nelze měnit interface zařízení ani využívat plnohodnotného multitaskingu (jedná se zde o modifikaci). Výrobce se zde snaží omezovat funkce, které dávají uživatelům určitou svobodu v nastavování zařízení, čímž dosahuje vysoké bezpečnosti, ale zároveň si tím vytváří odpůrce u některých uživatelů.

4.2.1 Uživatelská komunita systému iOS

Mimo samostatný vývoj má systém iOS aktivní hackerskou komunitu, která přinesla výzkum a nástroje pro obcházení systémových zabezpečení. Jedná se zde o zmiňovaný jailbreak, který odemkne zařízení a tudíž je do něj možné instalovat aplikace mimo internetový obchod Apple Store, což při normálním chodu zařízení není možné. Nejznámější aplikací umožňující jailbreak je *Cydia*. Tato aplikace se velmi proslavila díky snadnému odemknutí zařízení. I přesto, že jailbreak není silně podporován pro klasické uživatele už jen kvůli možnosti stahování placených aplikací zdarma, společnost Apple nezatracuje hackerskou komunitu, protože má určitý přínos. Jailbreak je velmi často využíván výrobci aplikací a to proto, že výrobci při testování nové aplikace nemusejí využívat složitých a zdlouhavých postupů pro otestování aplikace na zařízení pomocí SDK (Software Development Kit), ale mohou použít jailbreak pro rychlé nahrání aplikace a následný test. Další výhodou jailbreaku je mnohem lepší poznání samotného zařízení, protože umožňuje lepší poznání souborového systému a dalšího datového obsahu jako je například mobilní terminál **Chyba! Nenalezen zdroj odkazů.**.

4.2.2 Internetový obchod Apple Store

Všechna zařízení od společnosti Apple, která fungují na systému iOS, se dodávají se základní softwarovou výbavou (kalendář, galerie apod.). Avšak společnost Apple umožňuje výrobcům třetích stran vývoj libovolných aplikací, které si potom uživatelé systému iOS zařízení mohou stáhnout. Aby zde docílili bezpečnosti zařízení a vyhnuli se možnosti rozšíření škodlivého softwaru touto formou, vytvořili obchod s aplikacemi zvaný Apple Store. Apple Store je jedinou možnosti neodblokovaného zařízení (zařízení bez jailbreaku), jak do něj stáhnout a nainstalovat aplikace. Všechny aplikace předtím než jsou dostupné na Apple Store, jsou ověřovány a testovány firmou Apple, aby se zjistilo zda se nejedná o škodlivou aplikaci. Aplikace, které neprojdou testováním, nejsou na Apple Store poskytnuty a výrobci (žadateli o poskytnutí) je udán důvod proč. Tedy není zde možné dosáhnout toho, že si uživatel stáhne aplikaci, která bude fungovat jako škodlivý software. To systému iOS zajišťuje velmi vysokou ochranu proti stažení škodlivého softwaru **Chyba! Nenalezen zdroj odkazů.**

4.2.3 Počítačová aplikace iTunes

Softwarový program pro klasické počítače obsahující spoustu užitečných funkcí, které zajišťují bezpečnou komunikaci a přenos dat mezi počítačem a zařízením. Dnes již tento program není nutností k funkci zařízení jako u předchozích verzí systému iOS, které na něm byly silně závislé. I přesto tento program umožňuje uživateli používat funkce, které

pomohou lépe zabezpečit jeho zařízení.

První z funkcí, kterou aplikace iTunes nabízí, je synchronizace zařízení. Vzhledem k tomu, že z bezpečnostních důvodů není možné připojit zařízení k počítači jinak než přes aplikaci iTunes, všechny data, která chce uživatel přenést, musí projít přes tento program pomocí synchronizace zařízení. Tím se zajistí, že se do zařízení nedostanou nepodporované soubory. Na druhou stranu synchronizace znemožňuje použití zařízení jako přenosné úložiště dat. Další z funkcí je vytvoření zálohy zařízení. Program zde vytvoří zálohu všech dat obsažených v zařízení do počítače, čímž umožňuje při výskytu softwarové chyby v zařízení přehrání celého systému bez ztráty dat. S tím přichází další funkce, kterou je stahování aktualizací systému. Počítač připojený k internetové síti je schopný pomocí aplikace iTunes stáhnout aktualizaci systému (pokud je k dispozici) a bezpečně přeinstalovat zařízení beze ztráty uživatelských dat **Chyba! Nenalezen zdroj odkazů.**.

4.2.4 Služba iCloud

Služba iCloud je jedna z nových možností systému iOS. Umožňuje uživateli používat řadu rozsáhlých funkcí, od synchronizace uživatelských dat se vzdáleným serverem, až po funkci s názvem „*find my phone*“ (najdi můj telefon). Funkce *find my phone* je možné využít při ztrátě zařízení. Pokud je aktivována, může se uživatel při ztrátě zařízení přihlásit přes webovou stránku společnosti Apple na svůj účet a následně vyhledat přibližnou polohu svého zařízení pomocí systému GPS (Global Position System). Uživatel poté také může poslat zprávu na zařízení, která potvrdí, že on je majitelem a může žádat o navrácení mobilního zařízení. Pokud se však jedná o krádež, je zde také funkce, která zamkne zařízení na dálku kódem zadaným majitelem a je možné zvolit možnost vymazání všech dat ze zařízení **Chyba! Nenalezen zdroj odkazů.**.

4.2.5 Bezpečnostní architektura systému iOS

Vzhledem k možnostem zařízení fungujících na operačním systému iOS se u tohoto systému využívá aplikačních programovatelných rozhraní (dále jen API) pro tvorbu bezpečnosti. Tyto API jsou umístěny ve vrstvě služeb jádra (viz Obr. 4.3), kde k nim mohou přistupovat také výrobci aplikací. Model vrstev z hlediska bezpečnosti (viz Obr. 4.4) využívající API je popsán následovně.

Úschovna klíčů

V úschovně klíčů se ukládají všechna možná hesla, klíče, certifikáty a jiná tajemství. Avšak její realizace je závislá na dvou aspektech. Prvním jsou kryptografické funkce, které zde slouží k šifrování a dešifrování všech uvedených tajemstvích (hesla, klíče apod.). Druhým aspektem jsou funkce ukládání dat, protože uvedené tajemství musí být někde uloženy a stejně tak záznam, kde se dané klíče mohou používat. K dosažení těchto aspektů užívá úschovna klíčů dynamickou knihovnu Crypto.

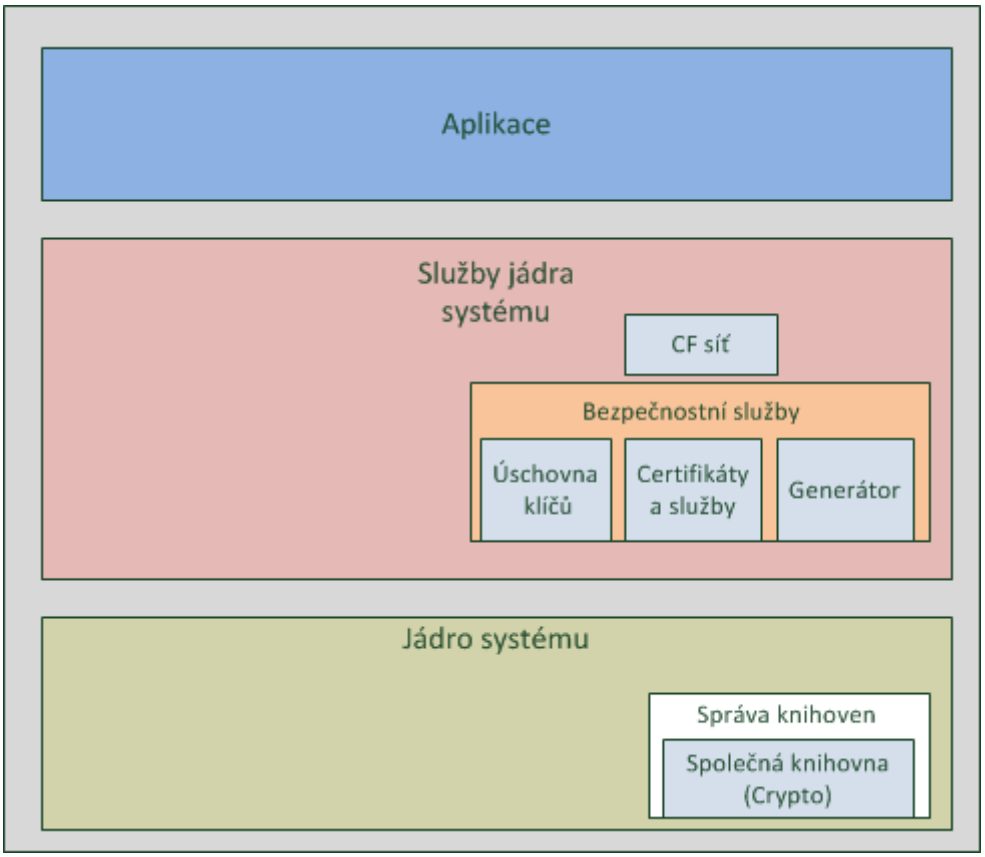
CF síť

Jedná se o vysokoúrovňovou API, kterou mohou aplikace používat k udržování

bezpečných datových toků a přidávání ověřovacích informací do různých zpráv. Pro svoji správnou funkčnost využívá všech položek uvedených v bezpečnostních službách (viz Obr. 4.4), tedy úschovny klíčů, generátoru, certifikátů a služeb.

Generátor

Generátor poskytuje kryptograficky bezpečná pseudonáhodná čísla. O generování čísel se stará algoritmus, který je volaný z vrstvy jádra systému. I přesto, že čísla nejsou čistě náhodná, není z nich možné odhalit algoritmus.



Obr. 4.4: Model vrstev operačního systému iOS z hlediska bezpečnosti [10].

Certifikáty a služby

Tato API je v podstatě sada funkcí, které se používají pro navázání bezpečné komunikace. Seznam funkcí je následující:

- Vytváření, správa a čtení certifikátů
- Přidávání certifikátů do úschovny klíčů
- Vytváření šifrovacích klíčů
- Šifrování a dešifrování dat
- Ověřování podpisů
- Zpráva zabezpečovací polity

K ukládání dat pro všechny zmíněné funkce využívá API společnou knihovnu

Crypto spolu s ostatními systémy obsaženými ve vrstvě jádra systému.

4.2.6 Příklady bezpečnostních chyb systému iOS

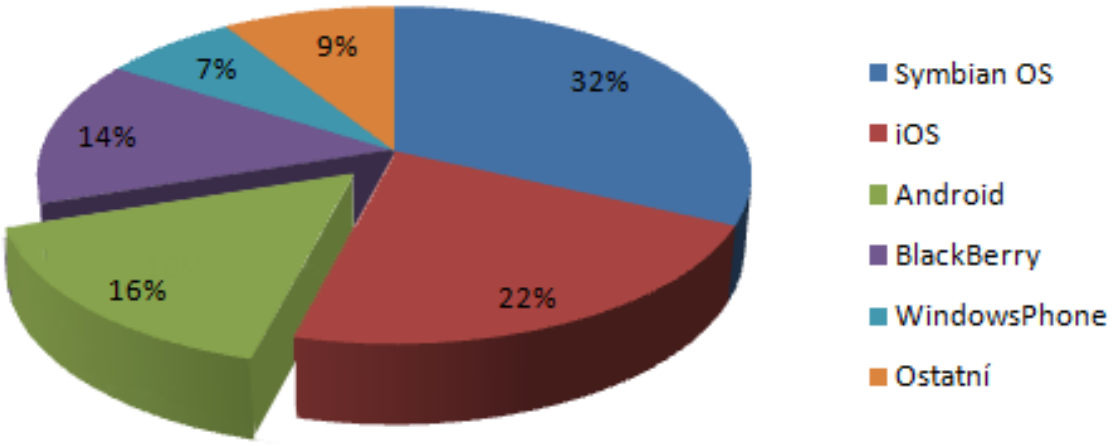
I přesto, že mobilní operační systém iOS je jeden z nejvíce bezpečných na trhu, bylo kvůli jeho popularitě nalezeno několik významných chyb, které silně narušovaly jeho bezpečnost.

Jednu z chyb v systému využili hackeři, kteří vytvořili exploit pro rozšíření viru, který lákal uživatele na odemčení telefonu. To se spíše týkalo amerického trhu, kde až donedávna nebylo možné použít kartu SIM (Subcriber Identity Module) od jiného operátora než u kterého byl telefon zakoupen. Po navštívení internetové stránky z mobilního telefonu, která zmíněné odemčení slibovala a po potvrzení odblokování telefonu se vymazala všechna data na zařízení a SIM kartě. Hackeři tohoto viru však neměli za cíl komunitu poškodit, ale spíše na chybu upozornit, protože se zde jednalo „pouze“ o smazání dat, které je možné obnovit a nedošlo k jejich odcizení. Společnost Apple na tuto chybu velice rychle zareagovala a vydala aktualizaci systému, kde již byla tato bezpečnostní trhlina opravena [12].

Další významnou chybou v systému byl virus zvaný SMS hack. Napadený uživatel tímto virem obdržel SMS na svoje mobilní zařízení, která obsahovala kód způsobující možnost ovládání zařízení útočníkem. Po úspěšném přijetí SMS již měl útočník plnou moc nad napadeným zařízením a mohl s ním provádět různé úkony od posílání SMS, přístupu k uživatelským datům apod. Tento virus vytvořil zabezpečovací expert Charlie Miller, který ho prezentoval na konferenci Black Hat, aby upozornil na závažnou chybu v kódu operačního systému. Tuto chybu však neobsahoval pouze operační systém iOS, ale Charlie Miller jí prezentoval i na ostatních operačních systémech. Dnes již je tato chyba opravena, ale uvádí se, že je jen otázkou času než někdo kód SMS útoku upraví tak, aby bylo možné útok provést znovu [13].

5. OPERAČNÍ SYSTÉM ANDROID

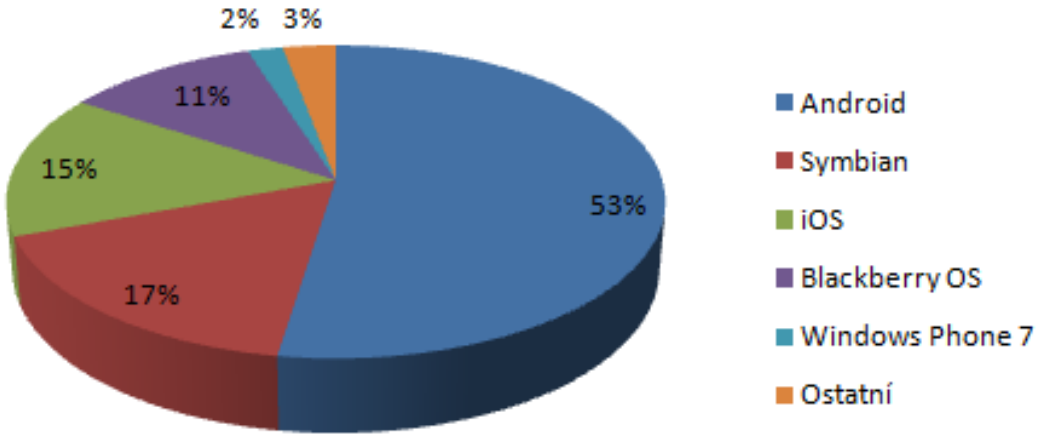
Android je otevřený (open-source) mobilní operační systém založený na jádře počítačového operačního systému Linux 2.6. Byl založený skupinou OHA (Open Handset Alliance), od které ho následně odkoupila společnost Google. První zařízení se systémem Android bylo vydáno v roce 2008, avšak přes relativně krátkou dobu působení na trhu se Android dokázal dostat mezi nejrozšířenější mobilní operační systémy s podílem 16% na celosvětovém trhu (viz Obr. 5.1).



Obr. 5.1: Podíl Androidu na celosvětovém trhu mobilních operačních systémů [1].

Jeho neuvěřitelná expanze na trhu je způsobena právě otevřeností kódu tohoto operačního systému. Google zde uvedl projekt AOSP (Android Open Source Project), čímž uvedl většinu zdrojového kódu jako bezplatnou softwarovou licenci *Apache*. To vedlo k zaměření výrobců právě na tento operační systém, protože sami nemusejí investovat velké peníze do vývoje vlastního operačního systému. Následkem toho je sice spousta zařízení fungujících na operačním systému Android, ale také spousta verzí tohoto systému. To by se nemuselo zdát nevýhodou za předpokladu, že by většina zařízení byla aktualizována na nejnovější verzi systému. Tak tomu však není, protože výrobci se musejí zabývat náročnou optimalizací. Místo toho vyrábějí zařízení nové s aktualizovaným systémem, aby zvýšili své tržby, což vede ke spoustě zařízení fungujících na starších verzích, kde jsou již odhaleny bezpečnostní chyby a existuje pro ně spousta škodlivého softwaru [14].

Obrovská popularita (viz Obr. 5.2), zmíněná otevřenost, ale také velké množství různých verzí systému vede k jeho špatné bezpečnosti. Nejde zde o nedostatečnou snahu výrobců systém zabezpečit, ale o velice lehké pochopení funkčnosti systému ze strany hackerů.



Obr. 5.2: Analýza prodeje chytrých telefonů za třetí kvartál roku 2011 na celosvětovém trhu [16].

5.1 Architektura systému Android

Android je operační systém fungující na jádře Linux 2.6. Jeho architektura obsahuje několik vrstev (viz Obr. 5.3), aby bylo docíleno co nejoptimálnějšího vývoje aplikací. Pro vývoj aplikací se zde používá nástroj Android SDK, které využívá nástrojů a API pro dosažení požadované funkčnosti a maximální bezpečnosti. Architektura se podle hlavních komponentů dělí následovně [14].

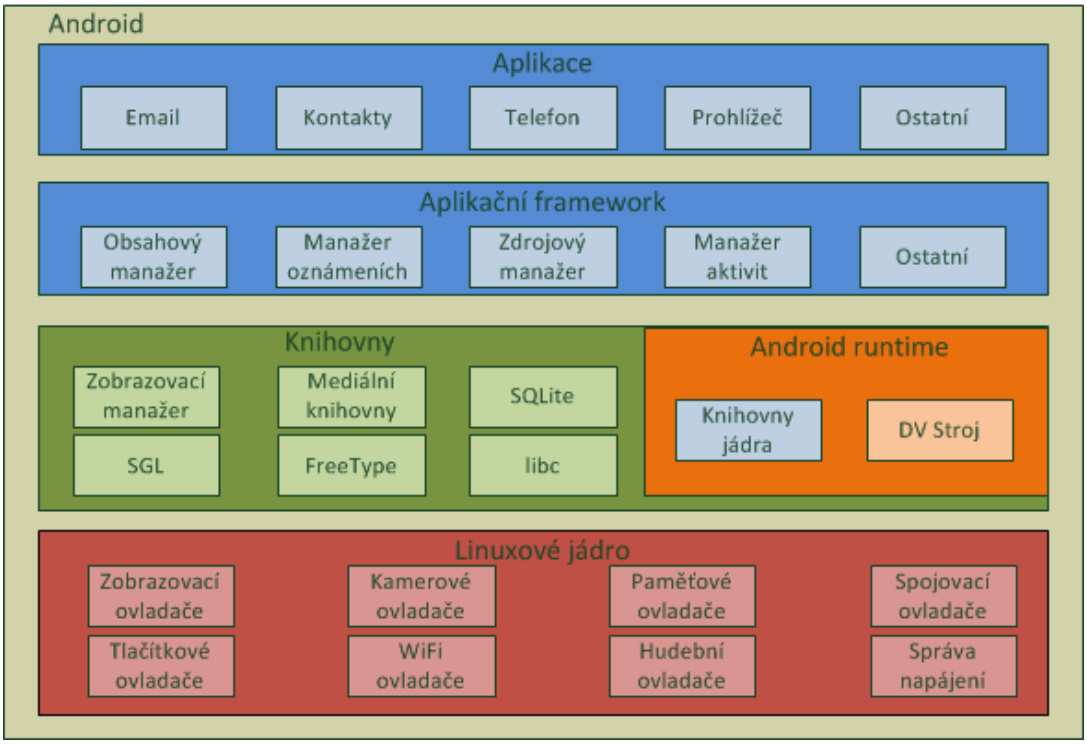
Aplikace

Nejvyšší vrstva obsahuje samotné aplikace. Mezi ně patří například emailový klient, kalendář, prohlížeč, kontakty, ale také aplikace třetích stran apod. Všechny aplikace psané pro Android se píší v programovacím jazyce Java.

Aplikační rámce

Vývojáři Androidu se snaží vybudovat velmi bohatý systém díky poskytování otevřené developerské platformy. Toho docilují pomocí poskytnutí plného přístupu k API rámcům, které používají funkce běžící v jádře. Díky tomu jakákoliv aplikace může používat všechny dostupné mechanismy (upozornění, lišta apod.) poskytované systémem Android. Mezi API poskytované na této vrstvě patří:

- *Obsahový manažer*, jehož úkolem je sdílení dat mezi různými aplikacemi.
- *Manažer oznámení* umožňující zobrazování alarmů v notifikační liště.
- *Zdrojový manažer*, který poskytuje přístup k nekódovaným zdrojům.
- *Manažer aktivit* starající se o životnost aplikací.



Obr. 5.3: Model vrstev operačního systému Android [15].

Knihovny

Android obsahuje sadu C/C++ knihoven používanou pro řadu různých komponentů. Funkce systému jsou pro vývojáře uloženy v této vrstvě a jsou dostupné z aplikačního rámce.

Android runtime

Tato vrstva obsahuje knihovny, které poskytují základní funkce jádra programovacího jazyku Java. Dále je zde tzv. Dalvikův virtuální stroj, který zajišťuje možnost multitaskingu u operačního systému Android.

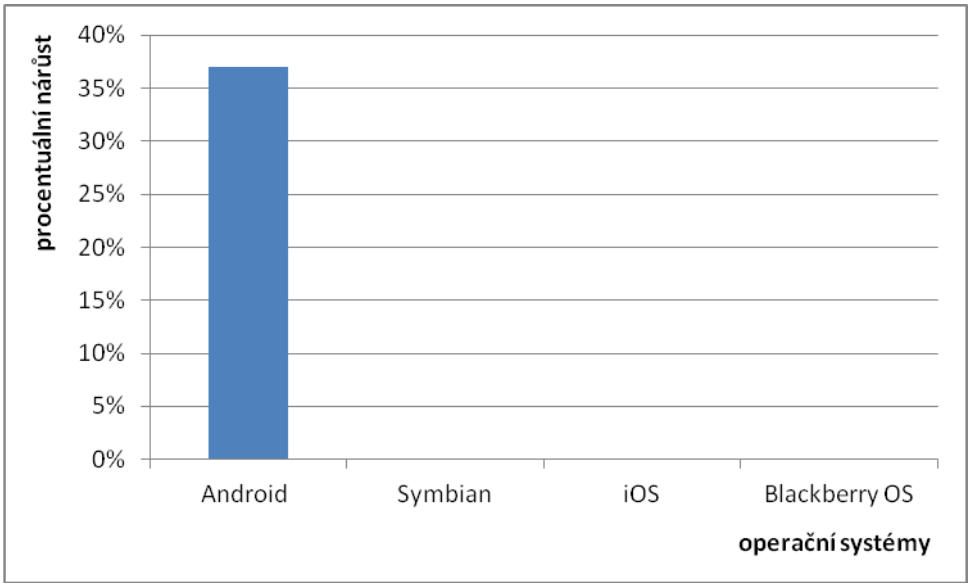
Linuxové jádro

Android funguje na Linuxovém jádře verze 2.6, které poskytuje služby jako zabezpečení, správu paměti, správu procesů a ovladače zařízení. Jádro také slouží jako vrstva zprostředkovávající komunikaci mezi softwarem a hardwarem.

5.2 Zabezpeční systému Android

Open source platforma Android od společnosti Google si za relativně krátkou dobu dokázala vydobýt na trhu chytrých telefonů velmi významnou pozici. I přesto, že je rozšířená mezi klasickými uživateli, nedaří se jí adaptace na podnikovou sféru. Na vině jsou chybějící funkce, ale zejména nedostatečné zabezpečení a správa. Androidu chybí zejména podpora šifrování dat, včetně enkrypce SD karet a také lepší ochranný systém proti mobilnímu malwaru. Ten se v poslední době zaměřuje na Android ve velkém.

Podle společnosti McAfee (antivirový výrobce), která sleduje výskyt škodlivého softwaru na operačním systému Android, existuje již přes 75 milionů škodlivých kódů pro tento systém. Stává se tak jedním z nejvíce nebezpečných vzhledem k tomu, že díky jeho popularitě se prakticky všechen nový malware vyrábí právě pro Android (viz Obr. 5.4), pro ostatní systémy je podíl zanedbatelný. Mezi škodlivý software patří například trojské koně, které potají odesílají osobní informace a kradou peníze nebo software, který je schopen nahrávat hovory a poté je odesílat útočníkovi [18].



Obr. 5.4: Procentuální nárůst škodlivého softwaru za třetí kvartál roku 2011 [18].

Androidu neprospěla ani skutečnost, že většina mobilních zařízení používajících tento systém obsahuje chyby dovolující například reklamním společnostem krádež přihlašovacích údajů používaných k přístupu ke kalendáři, kontaktům apod. Tato chyba byla částečně opravena s verzí systému 2.3.4, avšak i tato verze umožňuje podobný útok například při synchronizaci se službou Picasa.

5.2.1 Obchod s aplikacemi Android Market

Stejně jako některá konkurence také operační systém Android nabízí internetový obchod pro stahování aplikací určených pro zařízení nazvaný Android Market. I přesto, že výrobci třetích stran, které aplikace na tento obchod nahrávají, musí mít všechny nahrané aplikace digitálně podepsané, aby je bylo možné zpětně určit, neprochází žádnou kontrolou, zda je aplikace vhodná či ne před nahráním na Android Market. Aplikace do obchodu nahrané se kontrolují společností Google zpětně a pokud se zjistí, že je aplikace závadná, je z obchodu odstraněna. To však nezamezuje tomu, aby si jí ještě předtím nemohl nikdo stáhnout.

To že internetový obchod Android Market není dostatečně bezpečný, se také v minulosti potvrdilo škodlivým softwarem nazývaným *DroidDream*. Tímto softwarem bylo infikováno více než 50 různých aplikací nabízených prostřednictvím zmiňovaného

obchodu. Kód po stažení do zařízení začal zjišťovat dvě unikátní čísla používaná v zařízeních – IMSI (International Mobile Subscriber Identity) a IMEI (International Mobile Equipment Identity), což mu umožnilo získání přístupu. Poté stáhl systémovou aplikaci, která nemohla být samotným uživatelem odinstalována (nedostatečná práva) a začal shromažďovat další informace, jako jsou model telefonu, používaný jazyk apod. [17].

5.2.2 Uživatelská komunita

Operační systém Android má velmi rozsáhlou komunitu mezi uživateli. Tato komunita se zaměřuje na vylepšování, zpřístupňování a umožňování nových funkcí, které se v základní verzi zařízení nevyskytují.

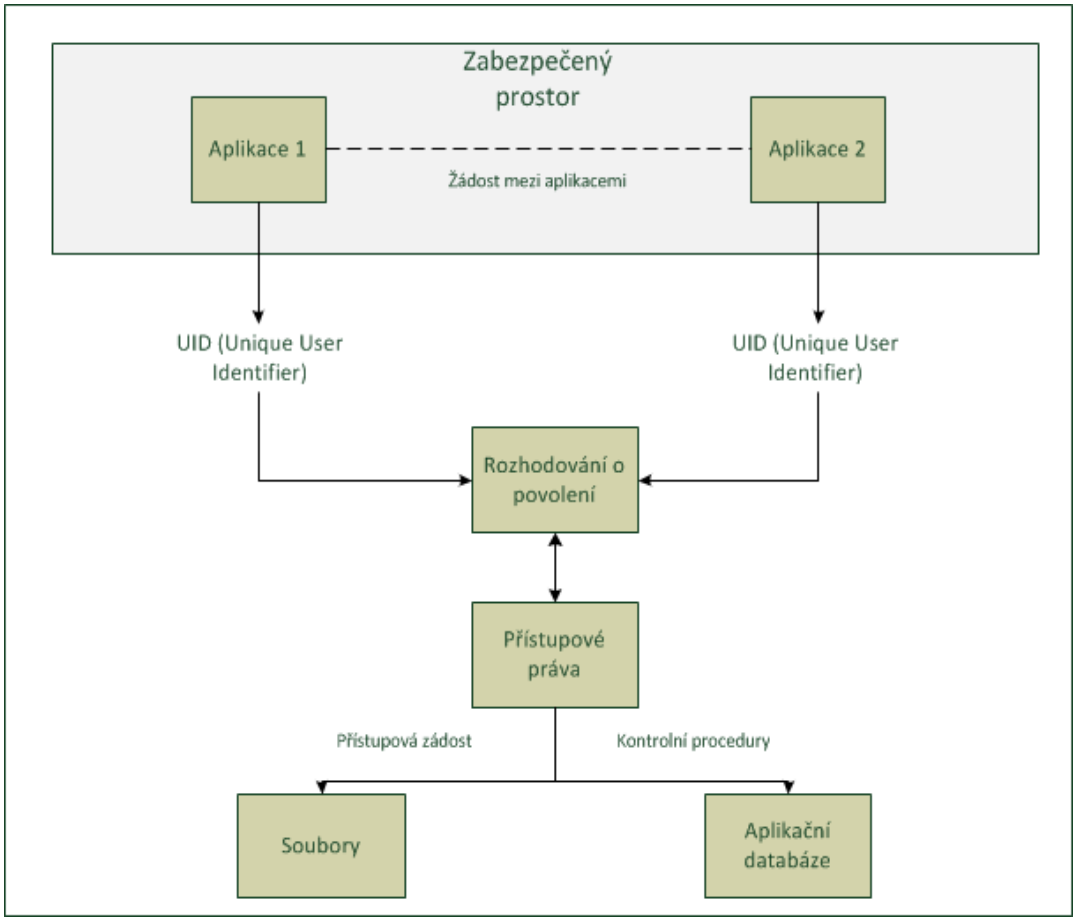
I přesto, že zásah do samotného zařízení přináší spoustu rizik, jsou nabízené možnosti vítány. Komunita poskytuje možnosti od odemčení zařízení (root) až po možnost nahrání operačního systému Android na zařízení, které pro něj není přímo určené nebo možnost nainstalování vyšší verze na starší zařízení. Všechny možnosti však zasahují do základního nastavení linoxého jádra, což může vést ke snadné chybě a následného vyřazení zařízení. Další nevýhodou je neúplná funkčnost těchto modifikovaných nástaveb, takže tak uživatel přichází o některé funkce zařízení.

5.2.3 Root zařízení

Root zařízení je internetový slangový výraz pro odemčení základního nastavení zařízení. Vzhledem k tomu, že je operační systém Android postaven na Linuxovém jádře a výrobce chce zde dosáhnout velké bezpečnosti, je přístup uživateli do nastavení tohoto jádra zakázán. Avšak díky zmiňované uživatelské komunitě je již možné pomocí spousty návodů na internetu toto uzamčení zařízení zrušit. To vede k možnosti základní nastavení zařízení změnit nebo dokonce možnosti přehrát operační systém v telefonu. Silně se to ovšem nedoporučuje, protože špatný zásah do tohoto nastavení může vést k celkovému kolapsu zařízení.

5.2.4 Bezpečnostní architektura

Android je více procesní systém, ve kterém má každá aplikace a součást systému vyhrazený svůj vlastní proces pro svoji funkčnost. Většina bezpečnostních opatření mezi aplikacemi a systémem jsou vynuceny skrz standardní Linoxové operace, mezi takové patří například UID (Unique User Identifier), které je pro každou aplikaci unikátní a slouží tak k jejich identifikaci. Další bezpečnostní opatření jsou prováděny pomocí mechanismu, který rozhoduje o povolení na specifické operace pro určitý proces. Tento mechanismus využívá přístupových práv.



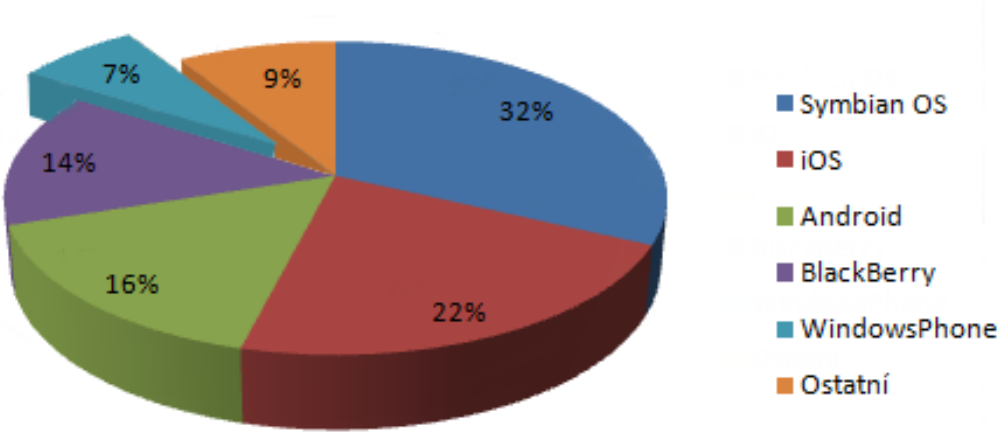
Obr. 5.5: Znázornění průběhů procesů v bezpečnostní architektuře Android OS [20].

Základním kamenem pro bezpečnostní architekturu operačního systému Android je, že žádná aplikace nemá povolení k provedení operace, která by nepříznivě ovlivnila aplikaci druhou, operační systém nebo uživatele. To zahrnuje čtení a zapisování uživatelských osobních dat (kontakty, email, apod.), čtení a zapisování dat jiných aplikací, přístup k síti, udržování zařízení v probuzeném režimu atd.

Díky zmíněným pravidlům se aplikace nachází v zabezpečeném prostoru, ve kterém nemůže být její chod narušen jinou aplikací kromě poslání žádosti mezi jednotlivými aplikacemi, která vyžaduje rozšířené možnosti, jež nejsou poskytnuty v základním zabezpečeném prostoru. Tyto žádosti se potvrzují nebo nepotvrzují na základě certifikátu a je nutné, aby byly staticky deklarované v dané aplikaci v době její instalace a poté je již nebylo možné změnit [20].

6. WINDOWS PHONE 7

Windows phone 7 je mobilní operační systém vytvořený společností Microsoft, kterého je možné považovat za nástupce operačního systému Windows mobile, i přesto že tyto systémy nejsou kompatibilní a nejsou si mezi sebou podobné. Windows phone je nejmladší na trhu a tudíž ještě není moc rozšířený. Z celosvětového trhu mobilních operačních systémů funguje na Windows phone 7 % zařízení (viz Obr. 6.1), což vzhledem ke konkurenci a krátké době na trhu není tak špatné, protože systém Windows phone 7 se zatím vyrábí pouze na mobilních zařízeních na rozdíl od ostatních mobilních operačních systémů.



Obr. 6.1: Podíl Windows phone 7 na celosvětovém trhu mobilních operačních systémů[1].

6.1 Zabezpečení systému Windows phone 7

Největší změnou u operačního systému Windows phone 7 oproti předchozímu produktu Windows mobile je zaměření na bezpečnost uživatele jak před úmyslným, tak neúmyslným poškozením. Bezpečnost „nových“ Windows phone 7 tedy zahrnuje prvky ochrany pro přenos dat, uložené data, stahování a instalování aplikací, ztrátu zařízení.

6.1.1 Ochrana přenosu dat

Zařízení s Windows phone 7 mají zabezpečený přenos dat pomocí ověřovacího mechanismu. Jedna se zde o využívání bezpečného spojení pomocí SSL (Secure Socket Layer) a digitálních certifikátů.

Kdykoliv, kdy je potřeba odeslat citlivá data, se využívá protokolu SSL, který je standardně určen pro šifrování dat. Tento protokol je řešením pro spojení mezi klientem a službou využívající asymetrické kryptografie a hierarchie certifikační autority (CA).

6.1.2 Ochrana uložených dat

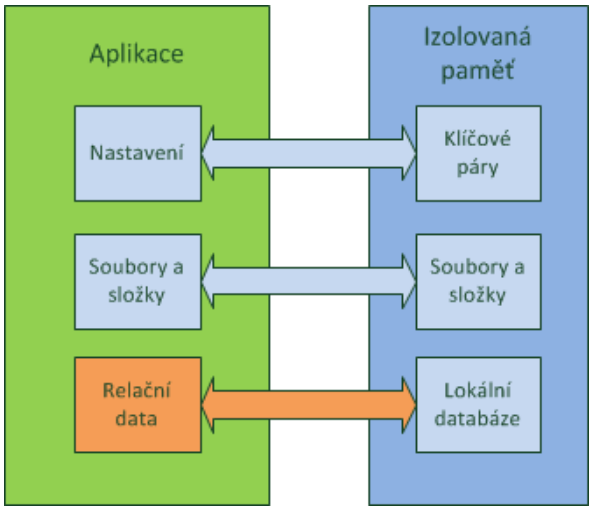
Ochrana uložených dat se u systému Windows phone 7 řeší pomocí modelu bezpečného prostředí. Tedy každé aplikaci se vytváří izolovaná paměť, do které si ukládá svoje data. Žádná aplikace nemá přístup k bezpečnému prostředí aplikace jiné, i přesto se zde ukládaná data šifrují, aby nebylo možné jich dosáhnout nikým jiným než samotným vlastníkem zařízení.

6.1.3 Stahování a instalování aplikací

Pro zachování vysoké bezpečnosti a snížení možnosti napadení zařízení škodlivým softwarem se u operačního systému Windows phone 7 pro instalaci aplikací zavedl internetový obchod zvaný Windows phone marketplace. V tomto obchodě se nacházejí všechny aplikace dostupné pro zařízení s daným operačním systémem a není možné aplikaci nainstalovat nějak jinak. Aplikace dodávané na marketplace výrobci třetích stran jsou přidány k dispozici pouze poté, co výrobce prokáže legitimní důkaz o identitě své společnosti a pouze pokud aplikace projde testovacím mechanismem, který má odhalit škodlivý software.

6.1.4 Bezpečnostní architektura

Bezpečnostní architektura u Windows phone 7 je založena na modelu vytváření bezpečného prostředí (viz Obr. 6.2) pro každou aplikaci poskytovanou výrobci třetích stran. Tedy žádná z těchto aplikací nemůže běžet na pozadí zařízení. Může přistupovat pouze k izolované paměti, nemůže přistupovat ke sdíleným datům systému a nemůže přímo přistupovat k uživatelským nebo telefonním funkcím.

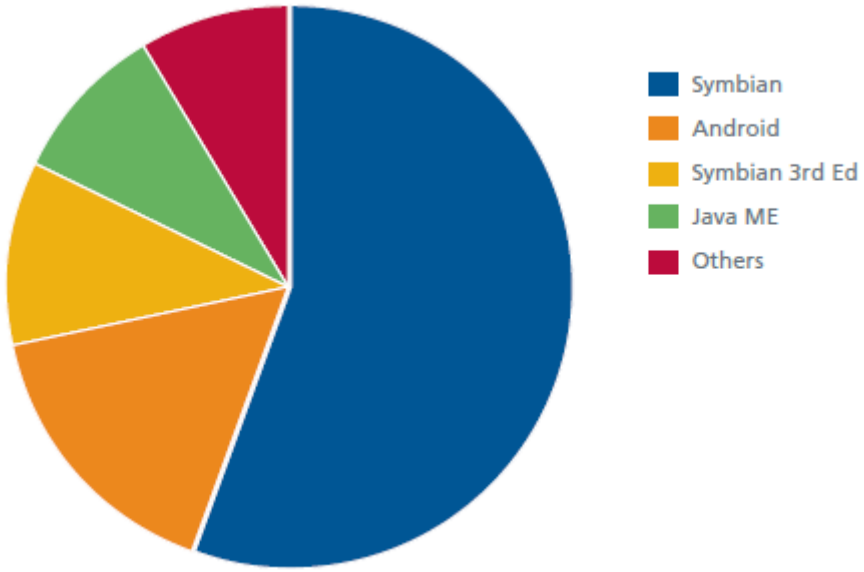


Obr. 6.2: Model bezpečného prostředí aplikace

Tento model bezpečného prostředí aplikace ukládá nastavení aplikace do izolované paměti pomocí šifrovaných klíčů a relační data do lokální databáze. Dále se zajišťuje vyšší bezpečnost nutností programování všech aplikací pomocí nástroje SDK.

7. POROVNÁNÍ ZABEZPEČENÍ MEZI JEDNOTLIVÝMI OS

Pro srovnání operačních systémů uvedených v této práci je nutné akceptovat dvě kritéria. Prvním kritériem je uvědomění všech možných bezpečnostních opatření, jako jsou například přístupnost systémového kódu, zabezpečení jednotlivých prvků apod. Druhým kritériem je rozšířenost operačního systému a množství škodlivého softwaru pro tento systém.

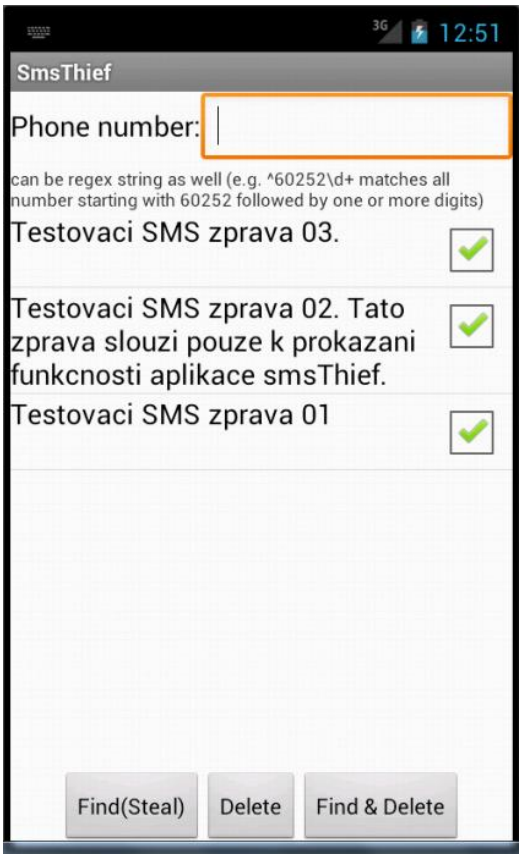


Obr. 7.1: Celkové zastoupení škodlivého softwaru pro jednotlivé platformy [18].

Při náhledu na statistiku rozšířenosti mobilního škodlivého softwaru pro jednotlivé systémy (viz Obr. 7.1) vychází Symbian OS jako nejméně bezpečný operační systém. To však není rozhodující, protože vzhledem ke grafu z předchozí kapitoly (viz Obr. 5.4) se nejvíce škodlivého softwaru v dnešní době objevuje pro operační systém Android. To ho velice jistě přesouvá na první místo mezi nejméně zabezpečenými mobilními operačními systémy. Následuje po něm, vzhledem k počtu škodlivého softwaru, operační systém Symbian OS. U zbylých dvou nelze tak jistě rozhodnout, vzhledem ke krátké době operačního systému Windows phone 7 na trhu. I přesto se tak nachází v žebříčku na nejnižší pozici díky vysoké bezpečnosti a skoro žádnému škodlivému softwaru v dnešní době.

8. ŠKODLIVÁ APLIKACE PRO OPERAČNÍ SYSTÉM ANDROID

V této kapitole je popsána tvorba a funkce na první pohled normální aplikace pro operační systém Android. Aplikace slouží jako jednoduchý správce SMS zpráv, který slouží pro vyhledávání a mazání přijatých SMS (viz Obr. 8.1). Zprávy se v telefonu vyhledávají na základě telefonního čísla odesílatele zprávy, které je možné zadat pouze částečně, celé nebo vůbec. Například pokud uživatel zadá do položky „Phone number“ číslici 7, vyhledají se všechny zprávy, které přišli od kontaktů, jejichž číslo začíná touto číslicí. Pokud uživatel zadá číslic více, vyhledají se kontakty obsahující tyto číslice v zadaném pořadí. Další možností je nezadání číslice žádné, v tom případě se vyhledají všechny SMS zprávy uložené v zařízení. Co však žádný uživatel, bez hlubšího prozkoumání nezjistí, je že aplikace při spuštění vyhledávání začne odesílat nalezené zprávy společně s číslem a kontaktem na vzdálený server pomocí internetového připojeni. Aplikace má pracovní název *smsThief*.



Obr. 8.1: Uživatelské prostředí aplikace *smsThief*.

8.1 Popis aplikace smsThief

Celková funkčnost aplikace jako SMS správce se škodlivým získáváním kontaktů a zpráv je rozdělena do sedmi tříd. Každá tato třída má určitou úlohu, která je nezbytná k plné funkčnosti aplikace. Rozdělení kódu do těchto tříd však není nutností, slouží pouze pro lepší pochopení zdrojového kódu a funkčnosti celé aplikace.

Třídy jako takové nejsou nijak rozděleny, avšak pro lepší pochopení aplikace se třídy dají rozdělit na ty, které slouží k funkčnosti správce SMS (**`ByNumArrayAdapter`**, **`ByNumDelActivity`**, **`SmsFindingTask`**) a na zbylé, které slouží ke krádeži určených dat (**`SmsThiefTask`**, **`Person`**, **`SmsMessage`**). Funkce jednotlivých tříd jsou následující.

8.1.1 Třída ByNumDelActivity

Je třída zahrnující aktivitu popisující vzhled původní obrazovky aplikace. Velkou součástí obrazovky je seznam (list) obsahující nalezené SMS po stisku tlačítka, který je definován adaptérem obsaženým v jiné třídě. Jsou zde definovaná jednotlivá obslužná tlačítka a jejich akce, které se vykonají v reakci na stisknutí. Příklad definice akce tlačítka vypadá následovně:

```
findBtn.setOnClickListener(new View.OnClickListener() {

    //@Override
    public void onClick(View v) {
      String regex =  regexInput.getText().toString();
      //starting finding task with hidden thief task
      new SmsFindingTask(ByNumDelActivity.this, regex,
      SmsFindingTask.ONLY_FIND, listOfFoundedSms, adapter).execute();

    }
});
```

Jedná se o tlačítko vyhledání (find), které po stisknutí zobrazí dialog informující uživatele o vyhledávání SMS a vyvolá tzv. asynchronní událost **`SmsFindingTask`**, kde se vyhledají zprávy odpovídající zadanému regulárnímu výrazu a pomocí adaptéru se zobrazí. Stisk ostatních tlačítek vyvolává jiné události, avšak jejich zdrojový kód je velmi podobný.

8.1.2 Třída ByNumArrayAdapter

Tato třída se zabývá vytvořením zmiňovaného seznamu (listu) s nalezenými zprávami, zobrazenými na obrazovce aplikace. Jedná se v podstatě o třídu, která po nalezení zprávy zobrazí její text do jednoho řádku, ke kterému navíc přiřadí checkbox, který slouží pro označování zpráv pro případné mazání. Vytváření řádků vypadá ve zdrojovém kódu následovně:

```
public View getView(int position, View convertView, ViewGroup parent){
      View view = null;
      //creating new "row"
      if (convertView == null) {
            LayoutInflater inflator = context.getLayoutInflater();
            view = inflator.inflate(R.layout.by_num_row_layout, null);
```

```
            final ViewHolder viewHolder = new ViewHolder();
            viewHolder.text = (TextView) view.findViewById(R.id.label);
            viewHolder.checkbox = (CheckBox)
view.findViewById(R.id.check);
            viewHolder.checkbox.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener()
```

8.1.3 Třída SmsFindingTask

Zde se jedná o třídu obstarávající vyhledávání jednotlivých zpráv. Existuje zde několik možností, v závislosti na zmáčknutém tlačítku, jak bude program dále pracovat. To je řešené přes přepínač, jehož část kódu je následující:

```
protected String doInBackground(String... params) {
      switch (taskType) {
      case ONLY_FIND:
            smsList.clear();
            findAndStoreByAddress();
            break;
```

V uvedeném případě se jedná o činnost prováděnou na pozadí v reakci na zmáčknutí tlačítka find. V tomto uvedeném případě se vymaže seznam dříve nalezených zpráv a zavolá se metoda označená `findAndStoreByAddress()`, která nalezne v telefonu hledané zprávy a uloží je do již vymazaného seznamu.

8.1.4 Třída SmsThiefTask

Až doposud se popsané třídy staraly o funkčnost aplikace z hlediska správce SMS zpráv. Tato třída se již však využívá pro škodlivou práci aplikace. Jedná se o třídu odesílající nalezená data na vzdálený server. Aby aplikace byla schopná odesílat nalezená data, musí třída obsahovat přesnou IP adresu vzdáleného serveru a port, které jsou definovány pomocí proměnných následovně:

```
      private static String hostIpAddress = "10.0.0.42";
      private static int hostPort = 8888;
      private Socket socket;
      private String terminatingLine = "Rowin commands - STOP\n";
```

Pro samotný přenos se využívají sockety. Klient serveru oznamuje ukončení přenosu dat zasláním oběma stranami domluveným řetězcem (`Rowin commands - STOP\n`). Jednotlivá data se odesílají jako objekty. To je také důvod proč musí aplikace a také server obsahovat shodné datové třídy (`Person`, `SmsMessage`) obsahující popis daných objektů.

Datové třídy `Person` a `SmsMessage` slouží k popsání objektů posílaných po síti. Pro odesílání SMS zprávy je v datové třídě `SmsMessage` definováno ID (identifikační číslo) zprávy, tělo zprávy, telefonní číslo odesilatele a datum přijetí. Pro kontakt je v datové třídě `Person` definováno ID kontaktu, jméno a číslo.

Celkový proces odesílání dat probíhá na pozadí a uživatel nemá ponětí, že k němu dochází. Pokud při odesílání nastane chyba, která způsobí nemožnost odeslání dat na server, jako například neaktivní internetové připojení, uživatel se to nijak nedozví. Aplikace nezobrazí žádné chybové hlášení ani neukončí svojí činnost.

8.1.5 TCPServer

Jedná se o jednoduchý TCP (Transmission Control Protocol) server, který slouží pouze k prokázání funkčnosti uvedené aplikace. Server poslouchá na všech síťových rozhraních na portu definovaném přímo v kódu. Tento server je vytvořen na základě návrhového vzoru Worker-farmer.

Pro demonstraci funkčnosti aplikace z hlediska odcizování zpráv a kontaktů bylo na zařízení posláno několik testovacích SMS zpráv od kontaktů uložených v paměti telefonu. Po použití aplikace se zprávy společně s kontakty odeslaly na uvedený server, který data sice neukládá, ale zobrazuje je ve svém výpisu (viz Obr. 8.2). Aby to ale bylo možné, musí být server zapnutý po dobu používání aplikace.

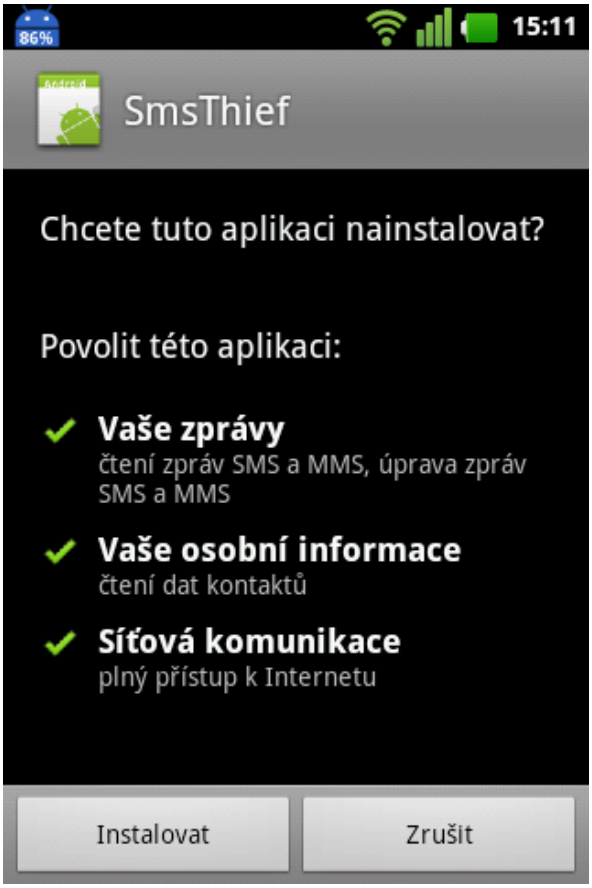
```
Server listen at port 8888
Connection from 127.0.0.1:52602
Begin contacts that SMS are from
================================
name: Ladislav Hromotny |       number: 987222111
name: Karel Matousek    |       number: 789111222

Begin stolen SMS listing
========================
From: 987222111 | when: Thu Apr 12 17:22:40 CEST 2012
 body: Testovaci SMS zprava 03.
From: 789111222 | when: Thu Apr 12 17:21:44 CEST 2012
 body: Testovaci SMS zprava 02. Tato zprava slouzi pouze k prokazani funkcnosti aplikace smsThief.
From: 789111222 | when: Thu Apr 12 17:21:06 CEST 2012
 body: Testovaci SMS zprava 01

Begin statistics
================
Contacts: 2     |       SMS: 3
Connection to client terminated...
```

Obr. 8.2: Výpis ze serveru obsahující odcizené data.

Všechny data přijatá na server se ve výpisu zobrazují v definovaném pořadí: jména kontaktů odcizených zpráv společně s telefonním číslem daného kontaktu, texty odcizených SMS zpráv s telefonním číslem odesílatele a datem přijetí a následně statistika zobrazující kolik kontaktů a zpráv bylo odcizeno.

8.2 Využitá slabina

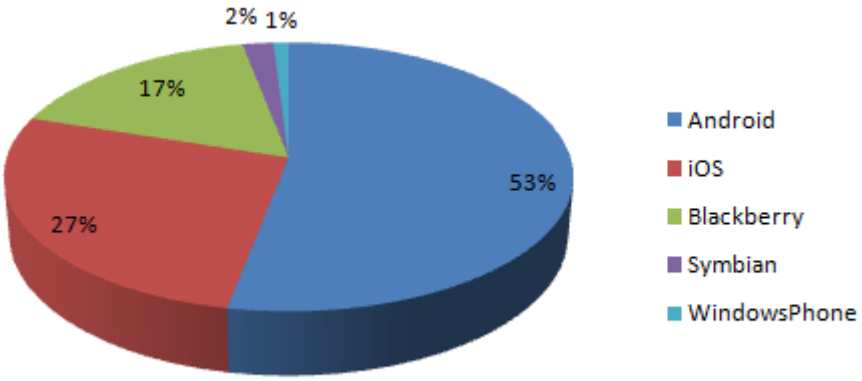
Každá aplikace pro operační systém Android musí využívat povolení pro přístup k určitým službám v tomto systému. Například pokud aplikace využívá přístup k SMS zprávám, musí být při vývoji zahrnuto povolení pro přístup k těmto zprávám. Ve finální podobě má každá aplikace seznam určitých povolení, které využívá. Uživatel při spuštění instalace aplikace musí potvrdit seznam všech povolení, které daná aplikace využije (viz Obr. 8.3), čímž souhlasí, že aplikace bude mít přístup k daným službám.



Obr. 8.3: Příklad seznamu povoleních při instalaci aplikace.

Tímto systémem povolení se v operačním systému Android zamezuje tomu, aby si uživatel nainstaloval aplikaci, která se bude zdát jako škodlivá. Avšak rozhodnutí, zda aplikace je škodlivá či není, na základě povolení, je pouze na daném uživateli, který nemusí mít žádné tušení, jak jsou povolení při instalaci důležité a bezmyšlenkovitě je odsouhlasí. Také se může vývojář aplikace vymluvit na využití povolení, které daná aplikace nepotřebuje. Typickým příkladem je vyžádání povolení k přístupu na internet za účelem stažení reklamy u bezplatných aplikací. Jak také uvádí statistika analytické společnosti Millennial Media za měsíc březen roku 2011, operační systém Android je lídrem v zobrazování reklam na mobilních zařízeních (viz Obr. 8.4).

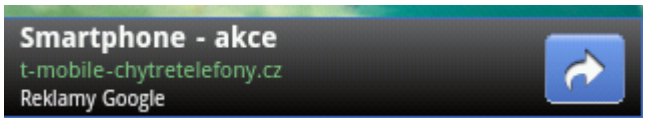
V bezpečném světě by tento fakt nebyl problém, avšak v tomto reálném světě tomu tak není. Jak již bylo zmiňováno v předchozí části této práce, hlavním zdrojem aplikací pro zařízení s operačním systémem Android je tzv. Android Market (nyní přejmenovaný na Google Play), na kterém nezávadnost aplikací není kontrolována. K odstranění škodlivých aplikací dochází pouze při nahlášení škodlivosti aplikace některým z uživatelů. To může vést k tomu, že některé aplikace shromažďují uživateli citlivá data, aniž by to kdy zjistil. Mezi taková citlivá data nemusí nutně patřit pouze SMS zprávy a kontakty. Jedná se zde o výpisy z bankovních účtů, přihlašovací údaje, ale také samotné hovory apod. Prakticky je možné sledovat jakoukoliv činnost na mobilním zařízení [23].



Obr. 8.4: Procentuální zastoupení počtu reklam pro jednotlivé OS [22].

Z různých zdrojů se informace o počtu škodlivých aplikací na Google Play liší. Například za březen roku 2011 bylo smazáno 50 škodlivých aplikací, které byly odhaleny. Tyto aplikace, předtím než byly odstraněny, si stáhlo přibližně 50 tisíc různých uživatelů. I přesto, že po odstranění aplikací z obchodu, byly také odstraněny na dálku ze zařízení, uživatelé tou dobou již byli infikováni škodlivým softwarem a mohli tak přijít o citlivá data.

Přiložená aplikace *smsThief*, je typickým příkladem využívající tuto slabinu. Jak je vidět ze seznamu povolení (viz Obr. 8.3), aplikace žádá o povolení číst uživatelovy zprávy, osobní informace a využívat síťovou komunikaci. Jak již bylo zmíněno, aplikace má sloužit jako správce SMS zpráv, tudíž by jí postačilo povolení ke zprávám uživatele a osobním informacím. Povolení pro síťovou komunikaci, které poskytuje plný přístup k internetu, je k funkčnosti samotného správce zpráv zbytečné a aplikace smsThief ji využívá pouze k odesílání ukradnutých dat na server. Avšak v tomto případě se dá, zde použité povolení, zamaskovat pomocí tzv. Google reklam (viz Obr. 8.5).



Obr. 8.5: Příklad typické Google reklamy.

Tyto reklamy, zobrazující se při používání aplikace, která je obsahuje, se stahují a přesměrovávají pomocí internetu, tudíž je nutné mít zahrnuto v seznamu povolení síťovou komunikaci. Co dalšího však aplikace s přístupem k danému povolení dělá, jde zjistit pouze po nahlédnutí do zdrojového kódu samotné aplikace, což je pro většinu uživatelů nepředstavitelné.

I přesto, že jsou Google reklamy způsob, jak zamaskovat škodlivé procesy v aplikacích, u přiložené aplikace *smsThief* tomu tak není. Je to dáno nutností založení Google účtu pro využívání těchto reklam, který je zpoplatněný a po případném odhalení škodlivosti aplikace by byl účet zablokován.

8.2.1 Metody šíření bez vědomí uživatele

V dnešní době je již zaznamenáno z minulosti několik velice sofistikovaných škodlivých aplikací, jejichž rozšíření nebylo závislé na Google Play. Tyto aplikace se jsou schopné šířit bez rizika, že je Google objeví, předtím než stihnou napáchat dostatečné škody. V takovýchto případech stačí, aby byl škodlivý software nainstalován pouze na několik zařízení, dále se poté šíří sám.

Příkladem takového softwaru je malware pojmenovaný UpdtBot. Tento škodlivý software se šíří pomocí SMS zpráv a to následujícím způsobem. Uživatel obdrží SMS zprávu, ve které je upozorněn na možnost instalace novější verze systému a varován, že pokud tak neudělá, hrozí mu vysoké riziko napadení zařízení vzhledem k nalezené slabině v systému. Zpráva také obsahuje odkaz na webovou stránku, která má danou aktualizaci obsahovat. Poté, co uživatel tuto stránku navštíví, se zařízení registruje a místo nainstalování zmiňované aktualizace se nainstaluje zmiňovaný UpdtBot, to je však nutné povolit dříve uváděným seznamem povolení (viz Obr. 8.3). Většina uživatelů přesto neověřenému zdroji uvěří a seznam potvrdí. V této fázi se již UpdtBot nainstaluje a umožní tak vzdálenému serveru kontrolu nad odesíláním zpráv, prováděním hovorů a instalaci dalšího softwaru. Podle uváděného zdroje byl UpdtBot rozšířen na více než 160 tisících zařízeních [24].

Většina podobných metod je vždy založena na oklamání uživatele, protože podle vývojářů operačního systému Android není možné, aby se jakýkoliv software nahrál do zařízení bez vědomí uživatele. Tedy vše co se na zařízení instaluje, je nutné potvrdit.

8.2.2 Ochrana

I přes všechny uvedené nedostatky v zabezpečení operačního systému Android je možné předejít útokům a jako uživatel se tak ochránit před hrozbou nahrání škodlivého softwaru. Postup pro zajištění bezpečnosti na mobilním zařízení je následující:

- V první řadě by uživatel měl stahovat aplikace pouze z důvěryhodných zdrojů, jakými jsou například Google Play, Tegra Zone apod. Před stahováním vybrané aplikace by měl dbát na hodnocení a názory od ostatních uživatelů a také na informace poskytnuté vývojářem dané aplikace.
- V druhé řadě by si měl uživatel, před instalací aplikace, pečlivě pročíst seznam povolení (viz Obr. 8.3) a ujistit se, že aplikace využívá pouze povolení, které opravdu potřebuje.
- Dalším krokem je sledování podezřelého chování zařízení. Za podezřelé chování je například považováno odesílání a přijímání neznámých SMS zpráv, příliš vysoká platba účtu od mobilního operátora z neznámých důvodů, ale také zvýšená spotřeba baterie, která může být způsobena ovládáním zařízení ze vzdáleného zdroje.
- Posledním krokem k dosažení optimální bezpečnosti by mělo být opatření antivirového programu, který však je známý svojí funkčností. U některých antivirů pro Android je běžné, že jejich funkcí je spíše oklamání uživatele za účelem reklamy, než ochrana zařízení. Mezi spolehlivé antiviry patří například McAfee WaveSecure nebo NQ Mobile Security.

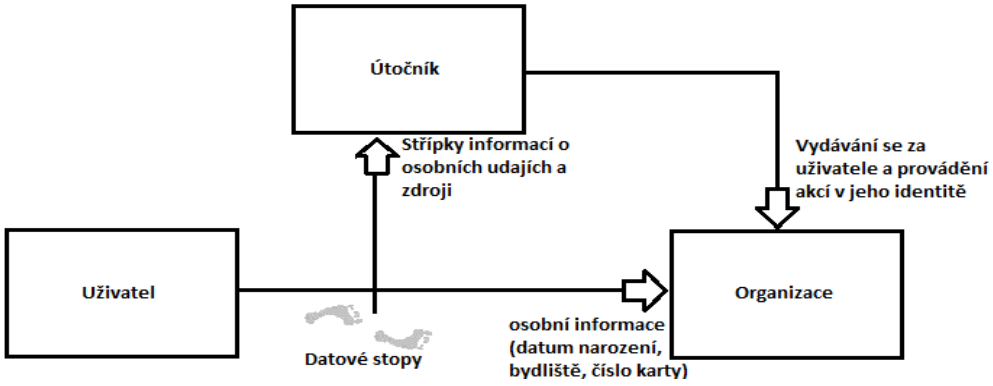
9. ZACHYTÁVÁNÍ DAT OPERAČNÍHO SYSTÉMU ANDROID

9.1 Cíl laboratorní úlohy

Cílem laboratorní úlohy je seznámení studentů s mobilním operačním systémem tzv. chytrých telefonů Android a následné poukázání na možné bezpečnostní chyby, které obsahuje. Studenti poznají škodlivou aplikaci využívající bezpečnostní slabiny, také se seznámí s možnostmi zabezpečeného a nezabezpečeného přenosu. Důvodem poukázání na zabezpečený přenos u aplikace pro sociální síť *facebook* je její nedávná absence šifrování dat. Tuto absenci šifrování dat má také aplikace *smsThief*, proto je použita v této laboratorní úloze, aby bylo názorné, jak je možné nešifrovaná data zachytit a následně přečíst. *Facebook* jakožto sociální síť může obsahovat citlivá osobní data, která při odcizení mohou způsobit uživateli újmu.

9.2 Úvod do problematiky

Operační systém Android se v dnešních dnech stal velice populární platformou pro chytré mobilní zařízení. I přes jeho velkou rozšířenost však není dostatečně zabezpečen, a proto se na něj objevuje stále více škodlivého softwaru tzv. malware. Za předpokladu, že uživatel je obezřetný, tedy stahuje aplikace pouze z ověřených zdrojů, pozorně sleduje, jak se aplikace chová a jaké má uživatelské hodnocení, zde však existuje ještě další riziko ztráty uživatelských dat. Tímto rizikem jsou různé aplikace, které nejsou škodlivé, ale jejich data odesílaná na internet nejsou dostatečně zabezpečena. Takovéto nedostatečné zabezpečení může vést k odcizení uživatelových citlivých dat pomocí zachytávání datových stop (viz Obr. 9.1) a důsledné ztráty určitého soukromí.



Obr. 9.1: Příklad odcizení uživatelských dat [26].

Operační systém Android zvládá širokou škálu komunikačních protokolů, které podporují zabezpečení před podobnými útoky při komunikaci na internetu. I přesto některé aplikace nevyužívají tyto možnosti a používají protokoly nezabezpečené.

9.2.1 Komunikační protokoly

Komunikační protokol je systém, podle kterého probíhá elektronická komunikace a přenos dat mezi dvěma koncovými body. Jedná se o definici pravidel, podle kterých se řídí, jak bude vzájemná komunikace probíhat a to z hlediska synchronizace dat, syntaxe a sémantiky. Různé protokoly mohou obsahovat signalizaci, ověřování doručení a detekci či případnou korekci chyb.

V rámci internetové komunikace se nejvíce používají protokoly modelu TCP/IP mezi které patří například IP (Internet Protocol), TCP, UDP (User Datagram Protocol) apod. Další velice často užívané protokoly pro komunikaci na internetu jsou HTTP (Hyper Text Transfer Protocol), DHCP (Dynamic Host Configuration Protocol), FTP (File Transfer Protocol), Telnet (Telecommunication Network) apod. Každý zmiňovaný protokol má jiné chování, zabezpečení a pravidla pro komunikaci a tím se hodí více či méně pro určitý druh komunikace [25].

Vhodným příkladem pro tuto laboratorní úlohu je protokol HTTP, který se používá pro výměnu hypertextových dokumentů, tedy slouží například pro zobrazování webových stránek. Jeho hlavní nevýhodou však je, že neumožňuje šifrování ani zabezpečení integrity dat. To znamená, že data, která jsou přeposílána mezi internetovým serverem a zařízením je možné zachytit a odhalit jejich obsah. To ovšem není problém pro většinu webových stránek vzhledem k tomu, že veškerá přeposílaná data není potřeba šifrovat. Obsah je volně dostupný a zachycená data nenesou žádnou podstatnou informaci o uživateli.

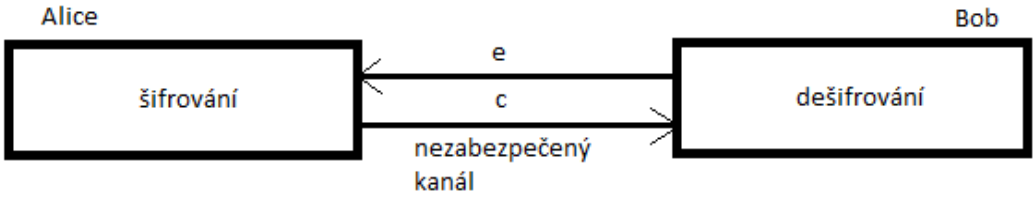
Na druhou stranu některé webové stránky vyžadují přihlášení a mohou obsahovat data, která uživatel může považovat za soukromá. V těchto případech by byl zmiňovaný protokol HTTP nevhodný, proto se používá jeho nástavba HTTPS (Hyper Transfer Protocol Secure). Jedná se o protokol využívající asymetrické šifrování pomocí SSL (Secure Socket Layer) nebo TLS (Transport Layer Security).

9.2.2 Šifrování dat

Šifrování dat se v dnešní době používá k zachování bezpečnosti v rámci internetu. Jedná se o určitý druh kryptografie, kde se pomocí matematických technik převádí informace do podoby, ze které je čitelná pouze se speciální znalostí. Kryptografie je velice obsáhlá a složitá nauka, proto pro danou úlohu bude popsán jen její základ a příslušné prvky sloužící pro šifrování dat na internetu.

Kryptografie jako taková se základně dělí na klasickou, symetrickou a asymetrickou. Pro účely této práce je důležitá pouze asymetrická kryptografie z hlediska jejího využití v zabezpečených komunikačních protokolech.

Asymetrické kryptografické systémy jsou založeny na použití dvou klíčů, pomocí kterých se šifrují přenášená data (zprávy). První klíč zvaný jako veřejný klíč slouží k zašifrování zprávy a je volně přístupný. Druhý klíč zvaný soukromý je vždy utajen a to pouze pro příjemce zprávy. Pro lepší pochopení je možné danou problematiku uvést na příkladě (viz Obr. 9.2).



Obr. 9.2: Příklad asymetrické komunikace [26].

Uživatel Alice chce poslat zabezpečenou zprávu uživateli Bob. Bob proto pošle Alici veřejný klíč *e* po nezabezpečeném kanálu. Alice pomocí zmiňovaného veřejného klíče *e* zašifruje svoji zprávu a tuto zašifrovanou zprávu *c* pošle Bobovi, který jí pomocí svého tajného klíče může dešifrovat. Zašifrovanou zprávu *c* není možné dešifrovat bez Bobova tajného klíče, proto se jedná o zabezpečený přenos [26].

Šifrování založené na tomto systému se používá například pro systémy SSL nebo TLS, které se starají o zabezpečení protokolu HTTPS. SSL je protokol, který se chová jako mezivrstva vložená mezi transportní a aplikační vrstvu. Funguje na principu uvedené asymetrické šifry, tedy každá z komunikujících stran má dvojici šifrovacích klíčů (veřejný a soukromý), které slouží k šifrování a dešifrování přeposílaných dat. TLS protokol je jeho nástavbou obsahující rozšiřující prvky pro komunikaci.

9.2.3 Wireshark

V této laboratorní úloze se pro zjištění, zda přenášená data aplikací jsou zašifrována při přenosu, použije program *Wireshark*. *Wireshark* je světově proslulý síťový protokolový analyzátor, který umožňuje zachytávání a interaktivní procházení provozu na počítačové síti. Program jako takový obsahuje velké množství funkcí. Pro tuto úlohu však postačí základní, které poslouží k zachytávání paketů odesílaných ze zařízení s operačním systémem Android na internet.

Pro tuto laboratorní úlohu postačí zachytávání paketů na zvoleném síťovém rozhraní, využití filtrů a zobrazování jednotlivých paketů s detailními informacemi mezi které patří například IP adresa zdroje a cíle, typ přenášených dat, práva, velikost hlavičky, typ použitého protokolu a v případě nešifrovaného přenosu i obsah. Pro lepší orientaci příklad zachyceného paketu vypadá následovně:

No.	Time	Source	Destination	Protocol	Length	Info
2314	4.985905	147.229.64.144	147.229.65.137	TCP	104	[TCP

Obr. 9.3: Příklad zachyceného paketu v programu *Wireshark*.

9.3 Postup

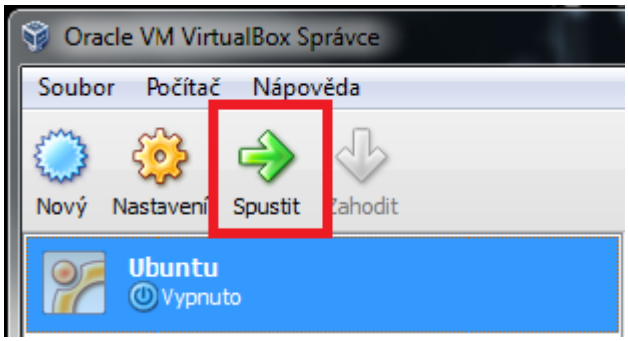
Před začátkem měření je nutné zkontrolovat příslušné vybavení. Pracoviště by mělo obsahovat mobilní telefon Samsung Galaxy Nexus, který, pokud není, zapněte a připojte k příslušné bezdrátové síti určené pro úlohu. Připojení k wifi se na zařízení

nachází přímo na pracovní ploše pomocí widgetu (zástupce) nebo v seznamu aplikací v položce nastavení – možnosti připojení. Připojení se indikuje v notifikační liště v horní části obrazovky, tuto lištu je možné otevřít pomocí táhlého pohybu z horní části obrazovky směrem dolů. Zařízení by mělo pro úlohu obsahovat aplikace *smsThief* a *facebook*. *SmsThief* je škodlivá aplikace vytvořená pro prezentaci nedostatečného zabezpečení operačního systému Android. Jedná se o jednoduchého SMS správce, který na pozadí, bez vědomí uživatele, odesílá nalezené SMS zprávy společně s kontakty na server. V poslední části, před zahájením měření, je nutné zapnout počítač (pokud není), na kterém se bude sledování provádět.

1. Zapněte vývojové prostředí *eclipse* ze seznamu Package Explorer v levém sloupci. Vyberte položku „*server*“ a následně tento server zapněte například pomocí klávesové kombinace „*ctrl+F11*“. Že je server zapnut je možné poznat pomocí výpisu „*Server listen at port:8888*“ v konzoli (viz Obr. 9.4). Úlohou tohoto serveru je příjem odcizených SMS a kontaktů pomocí aplikace smsThief. Aplikace se serverem si přenášejí data pomocí protokolu TCP (nešifrovaný) a data se na serveru pouze zobrazují (neukládají se), proto je nutné aby server běžel po celou dobu měření.

Obr. 9.4: Zobrazení spuštěného serveru v prostředí *eclipse*.

2. Spusťte program pro běh virtuálního systému (Oracle VM VirtualBox), který najdete na ploše. Zde ze seznamu vyberte operační systém *Ubuntu* a pomocí tlačítka „*Spustit*“ ho zapněte (viz Obr. 9.5).



Obr. 9.5: Spouštění virtuálního operačního systému.

3. Po naběhnutí virtuálního operačního systému je nutné nainstalovat program *aircrack*, který umožňuje zapnout promiskuitní režim síťové karty pro sledování provozu probíhajícího na síti. Pro jeho instalaci zapněte terminál pomocí současného stisku kláves „*alt+F2*“ a do zobrazeného okna napiště příkaz „gnome-terminal“ a potvrďte. Pro nainstalování příslušného programu je nutné do terminálu zadat následující příkazy:

```
wget http://download.aircrack-ng.org/aircrack-ng-1.1.tar.gz
tar -zxvf aircrack-ng-1.1.tar.gz
cd aircrack-ng-1.1
make
make install
```

Jedná se o získání instalačního balíčku programu (**wget**) z příslušné webové adresy. Vzhledem k tomu, že se jedná o zabalený soubor, je nutné ho rozbalit pomocí příkazu **tar** s příslušným doplňkem **-zxvf**. Následuje přesunutí do adresáře s rozbalenými soubory pomocí **cd**. Příkazem **make** dojde ke kompilaci zdrojových kódů do binární, tedy spustitelné podoby, **make install** poté tyto binární kódy přesune do spustitelných adresářů a vytvoří konfigurační soubory.

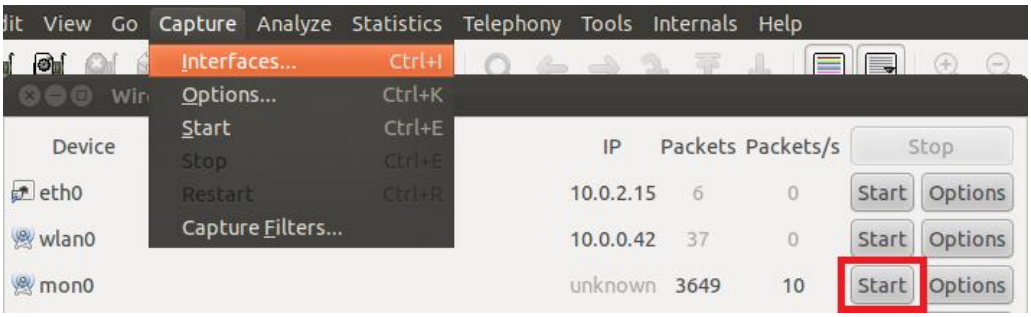
4. Nainstalovaný program spusťte pomocí příkazu „*sudo airmon-ng start wlan0*“, který následně zobrazí informace o síťovém rozhraní, které bylo upraveno (viz Obr. 9.6). Pomocí příkazu „*iwconfig*“ ověřte, že bylo vytvořeno nové rozhraní. Některé programy vyžadují přístup k funkcím systému, které nejsou pro klasické uživatele povoleny, proto se zde používá tzv. superuser, jež tyto práva má. Aby bylo možné se za zmiňovaného superusera přihlásit, používá se příkaz *sudo* a zadává se přístupové heslo.

```
Interface       Chipset         Driver

wlan0           Unknown         rt2800usb - [phy0]
                                (monitor mode enabled on mon0)
```

Obr. 9.6: Úprava síťového rozhraní pomocí programu *aircrack*.

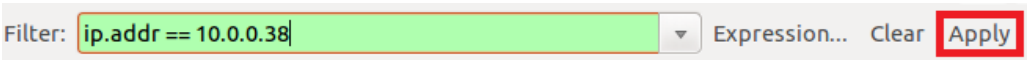
5. V následujícím kroku je nutné nainstalovat program *Wireshark*, který slouží pro sledování samotného provozu. Instalaci proveďte pomocí příkazu „*sudo apt-get install wireshark*“. Poté, co bude program nainstalován, ho spusťte pomocí „*sudo wireshark*“. V programu *Wireshark* se sleduje provoz síťového rozhraní, což je jedna z funkcí, které je možné využít pouze jako superuser.

6. V prostředí programu *Wireshark* spusťte zachytávání paketů, na Vámi vytvořeném rozhraní, pomocí *aircracku*. Pro zachytávání vyberte v horní liště programu položku *Capture* – *Interfaces*, vyberte rozhraní ze seznamu a stiskněte *Start* (viz Obr. 9.7).



Obr. 9.7: Spouštění zachytávání paketů v programu *Wireshark*.

7. Nyní na mobilním zařízení Samsung Galaxy Nexus spusťte aplikaci *smsThief*. Po spuštění aplikace Stiskněte tlačítko „*Find(Steal)*“. Aplikace Vám zobrazí všechny obsažené SMS zprávy v zařízení a tyto zprávy také odešle na server běžící v prostředí eclipse, tyto zprávy si přečtěte a ověřte jejich příjem na serveru (viz Obr. 8.2). Aplikaci vypněte a následně v *seznamu aplikací* v položce *nastavení – možnosti připojení* zjistěte IP adresu mobilního zařízení pomocí kliknutí na samotné připojení. Jedná se zde o jeden ze způsobů, jak zjistit IP adresu zařízení, aby pomocí ní bylo nadále možné vyhledávat konkrétní přenos.
8. Nyní se přepněte zpět do prostředí programu *Wireshark* a zastavte přenos (např. pomocí klávesové kombinace *ctrl+e*). V liště sloužící pro zadávání filtrů (označená nápisem *Filter:*) zadejte filtr, který Vám zobrazí komunikaci pouze mezi mobilním zařízením. Filtr je určen IP adresou mobilního zařízení zjištěnou v předchozím kroku. Příkaz filtru je následující „*ip.addr == "IP adresa zjištěná na zařízení"*“ (viz) .

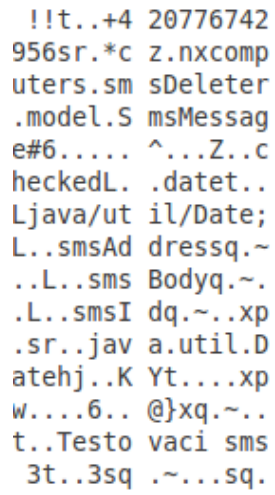


Obr. 9.8: Zadání filtru v programu *Wireshark*.

9. V zobrazeném seznamu paketů nalezněte komunikaci naposledy použité aplikace. K lepšímu nalezení může posloužit zjištění IP adresy serveru, který zprávy přijal (je možné použít pouze pro aplikaci *smsThief*). Po nalezení uvedených paketů zjistěte jejich protokol a zobrazte jejich obsah. Obsah paketu si nechte zkontrolovat vyučujícím.
10. Opakujte postup z bodu 4. a zvolte možnost pokračovat bez uložení. Na mobilním zařízení spusťte aplikaci pro sociální síť *facebook*. Aplikace je přihlášena na účet sloužící pro výuku, proto se z něj neodhlašujte! Aktualizujte prostředí aplikace pomocí přetažení hlavní části obrazovky směrem dolů, poté aplikaci vypněte. Opakujte postup uvedený v bodech 8. a 9. Zjistěte a odůvodněte rozdíly mezi přenosy použitých aplikací. Po skončení vypněte všechny spuštěné aplikace, jak na počítači, tak na mobilním zařízení.

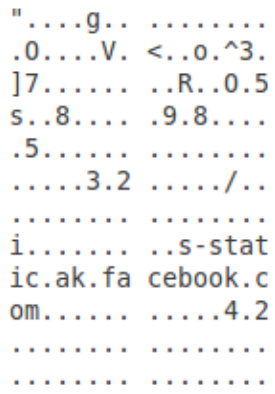
9.3.1 Výstup práce

Výstupem práce by mělo být zachycení dvou různých komunikací. V prvním případě jde o nešifrovanou komunikaci mezi aplikací *smsThief* a jejím serverem, která funguje bez vědomí uživatele. Studenti jsou s tímto faktem v průběhu úlohy seznámeni, a proto tento přenos zachytí. Jedná se o nešifrovaný přenos pomocí protokolu TCP, který obsahuje zprávy a kontakty odeslané z telefonu. Studenti pomocí programu *Wireshark* zachytí konkrétní pakety obsahující přenášené zprávy (viz Obr. 9.9) a tím si ověří možnou čitelnost nešifrovaného přenosu.



Obr. 9.9: Obsah zachyceného nešifrovaného paketu.

V druhém případě jde o šifrovanou komunikaci mezi aplikací pro sociální síť *facebook* a internetem. Zde studenti zachytí komunikaci mezi aplikací a vzdáleným serverem. Již se však jedná o šifrovaný přenos, proto ze zachycených paketů nebude možné vyčíst nic víc než identifikátor aplikace (viz Obr. 9.10).



Obr. 9.10: Obsah zachyceného šifrovaného paketu.

ZÁVĚR

Hlavním cílem této bakalářské práce bylo nastudovat a popsat dnešní mobilní operační systémy, zaměřit se na jejich zabezpečení, jejich porovnání a následné poukázání na nedostatečnou bezpečnost operačního systému Android. Rozebrány jsou čtyři v České republice nejznámější a nejpoužívanější mobilní operační systémy. Tedy Symbian, Android, iOS, Windows phone 7.

V první kapitole jsou popsány dnešní mobilní zařízení a slabiny, které byly postupem času odhaleny. Taktéž jsou v této kapitole zmíněny již pokročilejší útoky směřované právě na dnešní mobilní operační systémy.

V následující části práce jsou uvedeny způsoby prevence před dnes dostupným škodlivým softwarem. To se týká, jak seznámení uživatele s problematikou, tak volby vhodných programů pro zvýšení bezpečnosti.

Další čtyři kapitoly se již zabývají konkrétními operačními systémy, obecným seznámením s daným systémem a následně s architekturou a zabezpečením. Dále jsou tyto systémy porovnány z hlediska zabezpečení mezi sebou.

V osmé kapitole bakalářské práce je již popsána vytvořená aplikace a její server. Aplikace je škodlivým softwarem pro operační systém Android. Kapitola obsahuje základní popis, funkce a seznámení s kódem aplikace. Dále poté obsahuje vysvětlení využité slabiny, možnosti šíření škodlivých aplikací a návrh pro případnou ochranu před podobným softwarem.

Závěrečná kapitola je věnována navržené laboratorní úloze, jejíž úlohou je seznámit studenty s operačním systémem Android a jeho zabezpečením z hlediska používání komunikačních protokolů.

Součástí práce je vytvořená škodlivá aplikace pro operační systém Android a TCPserver potřebný pro funkci samotné aplikace. Aplikace je uložena ve formátu .apk pro případné nahrání přímo na zařízení a její kód je také přiložený a zabalený ve formátu .zip pro možnosti modifikací a nahlédnutí. Všechen, pro práci vytvořený software, byl vytvořen za pomocí programovacího jazyku java v prostředí programu eclipse.

LITERATURA

[1] StatCounter [online]. 2011 [cit. 2011-10-22]. Http://gs.statcounter.com. Dostupné z WWW: <http://gs.statcounter.com/#mobile_os-ww-monthly-201009-201109-bar>.

[2] DUNHAM, Ken. Mobile Malware Attacks and Defense. [s.l.]: Syngress Publishing, Inc, 2008. 440 s.ISBN 1597492981.

[3] Trifinite.org [online]. 2006 [cit. 2011-10-23]. Trifinite.stuff. Dostupné z WWW: <http://trifinite.org/trifinite_stuff.html>.

[4] GUO, Chuanxiong; WANG, Helen; ZHU, Wenwu. Microsoft research [online]. 2008 [cit. 2011-10-31]. Smart-Phone Attacks and Defenses. Dostupné z WWW: <http://research.microsoft.com/en-us/um/people/helenw/papers/smartphone.pdf>.

[5] COX, John. Networkworld [online]. 2009 [cit. 2011-10-31]. Mobile browsers bring new security headaches. Dostupné z WWW: <http://www.networkworld.com/news/2009/030409-mobile-browsers-security.html>.

[6] Nokia and Symbian OS. In White paper [online]. [s.l.]: Nokia, 2002 [cit. 2011-11-01]. Dostupné z WWW: <http://www.nokia.com/NOKIA_COM_1/About_Nokia/Press/White_Papers/pdf_files/symbian_net.pdf>.

[7] HEATH, Craig. Symbian OS Platform Security : Software Development Using the Symbian OS Security Architecture. [s.l.] : [s.n.], 2006. 276 s. ISBN 0470018828.

[8] Symbian Software Development. In ICT Communications & Multimedia [online]. Deventer: ICT Embedded B.V., 2007 [cit. 2011-11-02]. Dostupné z WWW: <http://www.ict.nl/C12572F4004BC1DB/files/Symbian%20Software%20Development%20-%20LowRes.pdf/$FILE/Symbian%20Software%20Development%20-%20LowRes.pdf>.

[9] Apple.com [online]. 01. 07. 2008 [cit. 2011-11-19]. Apple-iphone-Software Update. Dostupné z WWW: <http://www.apple.com/iphone/ios/>.

[10] Apple developer [online]. 12. 10. 2011 [cit. 2011-11-20]. IOS Technology Overview. Dostupné z WWW: <http://developer.apple.com/library/ios/documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/iPhoneOSTechOverview.pdf>.

[11] HOOG, Andrew; STRZEMPKA, Katie. IPhone and iOS Forensics : Investigation, Analysis and Mobile Security for Apple iPhone, iPad and iOS Devices. [s.1.]: Syngress, 2011. 336 s. ISBN 1597496596

[12] Iphonealley.com [online]. 21. 04. 2011 [cit. 2011-11-26]. IPhone Alley - iPhone news, app reviews, and accessories. Dostupné z WWW: <http://www.iphonealley.com/news/ios-virus-acquirable-through-safari-gaining-momentum>.

[13] Fuzzin the Phone in your Phone. Black Hat USA [online]. 25. 06. 2009, [cit. 2011-11-26]. Dostupný z WWW: <http://mashable.com/2009/07/30/iphone-hack/>.

[14] HOOG, Andrew. Android Forensics : Investigation, Analysis and Mobile Security for Google Android. [s.l.]: Syngress Publishing, Inc, 2011. 432 s. ISBN 1597496510.

[15] Android Forensics: Simplifying Cell Phone Examinations. Small Scale Digital Device Forensics Journal [online]. Září 2010, vol. 4, [cit. 2011-11-30]. Dostupný z WWW: <http://www.ssddfj.org/papers/SSDDFJ_V4_1_Lessard_Kessler.pdf>. ISSN 1941-6164.

[16] Gartner [online]. listopad 2011 [cit. 2011-11-27]. Gartner's analysis of global Q3 2011 smartphone sales. Dostupné z WWW: <http://www.gartner.com>.

[17] COMPUTERWORLD [online]. 08. 03. 2011 [cit. 2011-11-30]. Bezpečnost Android Marketu. Dostupné z WWW: <http://computerworld.cz/bezpecnost/google-zlepsuje-bezpecnost-android-marketu-42901>.

[18] McAfee An Intel Company [online]. 2011 [cit. 2011-11-30]. McAfee Threats Report: Third Quarter 2011. Dostupné z WWW: <http://www.mcafee.com/us/resources/reports/rp-quarterly-threat-q3-2011.pdf>.

[19] Android Developers [online]. 2011 [cit. 2011-11-30]. What is Android?. Dostupné z WWW: <http://developer.android.com/guide/basics/what-is-android.html>.

[20] JANTSCHER, Martin, et al. Mobile Application Development [online]. 2009 [cit. 2011-12-04]. Android team. Dostupné z WWW: <http://www.mad-ip.eu/files/reports/Android.pdf>.

[21] Msdn [online]. 23. 09. 2011 [cit. 2011-12-05]. Security for Windows Phone. Dostupné z WWW: <http://msdn.microsoft.com/en-us/library/ff402533(v=VS.92).aspx>.

[22] MILLENNIAL MEDIA. Millennial Media Releases: Mobile Mix Report [online]. 14. 04. 2011 [cit. 2012-04-13]. Dostupné z: http://www.millennialmedia.com/blog/2011/04/millennial-media-releases-march-mobile-mix-report/

[23] NQ MOBILE U.S. SECURITY RESEARCH CENTER. Security Alert: New Android Malware [online]. 05. 04. 2012 [cit. 2012-04-13]. Dostupné z: http://research.nq.com/?p=402

[24] NQ MOBILE U.S. SECURITY RESEARCH CENTER. Security Alert: New Android Malware — UpdtBot [online]. 11. 04. 2012 [cit. 2012-04-15]. Dostupné z: http://research.nq.com/?p=410

[25] DOSTÁLEK, Libor. Velký průvodce protokoly TCP/IP a systémem DNS [online]. 2. aktualiz. vyd. Praha: Computer Press, 2000, 426 s. [cit. 2012-04-23]. ISBN 80-722-6323-4.

[26] MENEZES, A., OORSCHOT, P., VANSTONE, S. Handbook of applied Cryptography. USA: CRC PRESS, 2001. 792 s. ISBN 0-8493-8523-7.

SEZNAM SYMBOLŮ A ZKRATEK

SMS	Short Message Service
OS	Operating System
DoS	Denial of Service
J2ME	Java 2 Platform, Micro Edition
PIN	Personal Identification Number
NFC	Near Field Communication
EKA2	EPOC Kernel Architecture 2
PIM	Personal Information Manager
US-CERT	United States Compoture Emergency Readiness Team
SDK	Software Development Kit
GPS	Global Position Systém
API	Application Programming Interface
SIM	Subcriber Identity Module
OHA	Open Handset Alliance
AOSP	Android Open Source Project
SD	Secure Digital
IMSI	International Mobile Subcriber Identity
IMEI	International Moboile Equipment Identity
UID	Unique User Identifier
SSL	Secure Socket Layer
ID	Identifier
TCP	Transmission Control Protocol
CA	Certification Authority
UDP	User Datagram Protocol
HTTP/S	Hypertext Transfer Procol / Secure
DHCP	Dynamic Host Configuration Protocol
FTP	File Transfer Protocol
TLS	Transport Layer Security