Detekce malwaru v mobilních zařízeních

Detection of Malware in Mobile Devices

Bc. Miroslav Palla

Diplomová práce
2014

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
akademický rok: 2013/2014

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Bc. Miroslav Palla**
Osobní číslo:	**A12350**
Studijní program:	**N3902 Inženýrská informatika**
Studijní obor:	**Bezpečnostní technologie, systémy a management**
Forma studia:	**kombinovaná**
Téma práce:	**Detekce malwaru v mobilních zařízeních**
Téma anglicky:	**The Detection of Malware in Mobile Devices**

Zásady pro vypracování:

1. **Prostudujte doporučenou literaturu a vypracujte literární rešerši na dané téma.**
2. **Popište základní architekturu mobilních operačních systémů iOS, Android a Windows Phone.**
3. **Proveďte analýzu implementovaných ochran před malwarem na jednotlivých platformách.**
4. **Proveďte test detekce malwaru na vybraných příkladech.**
5. **Navrhněte možnosti zabezpečení mobilních zařízení před jejich zneužitím a ochranou dat.**

Rozsah diplomové práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. **MILLER, Charles. IOS hacker's handbook. Indianapolis, IN: Wiley, 2012, 388 p. ISBN 11-182-0412-3.**
2. **DUNHAM, Ken. Mobile malware attacks and defense. Burlington, MA: Elsevier, 2009, 409 p. ISBN 15-974-9298-1.**
3. **HOOG, Andrew. Android forensics: investigation, analysis, and mobile security for Google Android. Amsterdam: Elsevier, 2011, 372 s. ISBN 978-1-59749-651-3.**
4. **HOOG, Andrew. IPhone and iOS forensics: investigation, analysis, and mobile security for Apple iPhone, iPad, and iOS devices. Amsterdam: Elsevier, 2011, 310 s. ISBN 978-1-59749-659-9.**
5. **SIX, Jeff. Application security for the Android platform. 1st ed. Sebastopol, CA: O'Reilly, 2012, 97 p. ISBN 14-493-1507-0.**
6. **HIMANSHU DWIVEDI, Chris Clark. Mobile application security. New York: McGraw-Hill, 2010. ISBN 978-007-1633-574.**

Vedoucí diplomové práce:	**Ing. David Malaník, Ph.D.** Ústav informatiky a umělé inteligence
Datum zadání diplomové práce:	**7. února 2014**
Termín odevzdání diplomové práce:	**27. května 2014**

Ve Zlíně dne 7. února 2014

prof. Ing. Vladimír Vašek, CSc.
děkan

UNIVERZITA TOMÁŠE BATI VE ZLÍNĚ
Fakulta aplikované informatiky
L.S.

doc. RNDr. Vojtěch Křesálek, CSc.
ředitel ústavu

ABSTRAKT

Předmětem diplomové práce je ochrana mobilních operačních systémů Android, iOS a Windows Phone 8. V teoretické části je popsána architektura těchto systémů a je zde provedena analýza bezpečnostních prvků, které tyto systémy chrání před malwarem. Praktická část je věnována především systému Android, kdy je proveden test detekce škodlivé aplikace ve virtuálním prostředí VirtualBox. V této kapitole je za pomocí obrázku ukázáno, jak tento systém na škodlivou aplikaci reaguje a jak si lze prostřednictvím vybraných webových služeb ověřit, zdali je aplikace opravdu škodlivá. V poslední části práce jsou navrženy možnosti zabezpečení mobilních zařízení, kde jsou uvedeny obecné zásady bezpečnosti a za pomocí speciálního softwaru je zde provedena praktická ukázka zabezpečení zařízení ve firemním prostředí.

Klíčová slova: Malware, Android, iOS, Windows Phone, Verify apps, VirusTotal, Anubis, VirtualBox, Avast, iPCU

ABSTRACT

This diploma work employs itself with a protection of mobile operational systems Android, iOS and Windows Phone 8. Architectures of these systems are described in the theoretical part as well as analyse of security elements, which secure these systems against malware. The practical part is dedicated mainly to Android system, when detection test of harmful application in a virtual environment named VirtualBox is being made. It is shown, how this system can react on the harmful application with the help of animation in this capture as well as how it is possible to check whether the application is being damaged, through a dedicated web services. The last part is dedicated to possibilities of mobile equipment securing, where general security policies are mentioned and a practical example of the equipment securing in a commercial environment is being made with help of a special software in this part.

Keywords: Malware, Android, iOS, Windows Phone, Verify apps, VirusTotal, Anubis, VirtualBox, Avast, iPCU

Tímto bych chtěl poděkovat Ing. Davidu Malaníkovi, Ph.D. za odborné vedení, cenné rady a připomínky při vypracování diplomové práce. Dále bych rád poděkoval své rodině a přítelkyni za poskytnutou psychickou podporu v období celého studia.

Prohlašuji, že

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

Prohlašuji,

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

Ve Zlíně …….

podpis diplomanta

OBSAH

ÚVOD

V současné době jsou mobilní zařízení pro stále více lidí naprostou samozřejmostí, ne-li nutností. Někteří lidé si dokonce bez těchto zařízení nedokáží svůj život ani představit. Mezi jejich největší výhody patří především jejich mobilita a možnost se prostřednictvím nich snadno dostat k potřebným informacím. Jako milník v této oblasti se dá považovat rok 2007, kdy společnost Apple Inc. představila svůj mobilní telefon iPhone, který disponoval dotykovou obrazovkou a na tu dobu pokročilým operačním systémem. Právě díky neustále se vyvíjejícím operačním systémům pro mobilní telefony došlo i k velkému rozmachu jednotlivých hrozeb, které si kladou za cíl právě tato zařízení. Zde útočníci využívají především neznalosti a nepozornosti samotných uživatelů a nejčastější ve formě škodlivých aplikací se z jejich zařízení snaží odcizit citlivé informace nebo vylákat finanční hotovost prostřednictvím prémiových SMS zpráv. Ne však každý uživatel si je vědom veškerých rizik, které z používání těchto zařízení vyplývají. V následující práci bude čtenářovi umožněno nahlédnout do problematiky používání mobilních zařízení se zaměřením na jejich operační systémy a také na bezpečnostní prvky, které tyto systémy chrání.

V teoretické části si řekneme, co to je malware, do jakých kategorií jej můžeme zařadit a jaká rizika jednotlivé kategorie představují. Rovněž si zde představíme některé zástupce malwaru, které se vztahují především k počátkům vývoje mobilních telefonů. V další části práce se budeme podrobněji zabývat operačními systémy, které jsou vyvíjeny speciálně pro mobilní zařízení. Budou zde popsány systémy Android, iOS a Windows Phone 8, jenž v současné době patří mezi nejpoužívanější zástupce těchto systémů. Všechny tři systémy jsou popsány z hlediska své historie, architektury a také jsou zde popsány bezpečnostní prvky, které tyto systémy chrání.

V praktické části je největší pozornost věnována systému Android, který je ze všech tří systémů nejvíce zranitelný. V této kapitole je za pomocí škodlivé aplikace otestována přítomnost funkce, která má za úkol detekovat potenciální nebezpečí. V tomto případě se jedná o kontrolu každé aplikace, kterou se uživatel chystá do svého zařízení nainstalovat. K tomuto účelu je využito virtuálního prostředí VirtualBox. Jakmile bude provedena detekce přítomnosti škodlivé aplikace, bude čtenáři ukázáno, jak lze za pomocí vybraných webových služeb tuto aplikaci analyzovat a identifikovat tak hrozbu, kterou představuje. V této části je rozebrána především problematika některých oprávnění, které aplikace pro

svoji funkci vyžadují a zároveň které mohou být pro uživatele prvním varováním, že se jedná o škodlivou aplikaci. Dále je v této kapitole vysvětlen bezpečnostní spouštěcí proces, který představuje základní bezpečnostní prvek ochrany systému iOS.

V poslední kapitole jsou navrženy možnosti zabezpečení mobilních zařízení, kde je formou obecných bezpečnostních zásad apelováno především na jednotlivé uživatele, aby si nebezpečí vyplývající z používání svých zařízení uvědomili, a využívali tak bezpečnostní prvky, které jednotlivé systémy nabízí. Právě využívání těchto bezpečnostních prvků představuje základní opatření, které značně snižuje riziko zneužití informací v případě ztráty nebo odcizení zařízení. Dále jsou v této kapitole představeny některé možnosti ochrany zařízení prostřednictvím aplikací třetích stran. Zde je názorně ukázáno na systému Android, jak lze využívat antivirové produkty, které v současné době nechrání systém pouze proti škodlivým programům, ale dokáží využívat i řadu dodatečných funkcí, jako je např. zálohování nebo lokalizace zařízení na dálku. V závěru práce je věnována zvýšená pozornost v oblasti firemních zařízení, kde je upozorněno na možné nedostatky, které se při používání těchto zařízení mohou vyskytovat. V této části je také čtenáři ukázáno na systému iOS, jak lze za použití speciálního softwaru chránit zařízení s tímto operačním systémem.

I. TEORETICKÁ ČÁST

1 MOBILNÍ MALWARE

Slovo Malware někdy také označováno jako Malicious Software (škodlivý software) je označením pro jakýkoliv program, který dělá něco, co způsobuje škodu uživateli, počítači nebo síti [20]. V poslední době s příchodem chytrých telefonů a jiných mobilních zařízení se stále více hovoří o tzv. mobilním malwaru. Jedná se o škodlivé programy, které si za svůj cíl vybírají mobilní telefony, tablety nebo jiná smartphone zařízení. Většina těchto programů je navržena tak, aby útočník mohl převzít kontrolu nad napadeným zařízením nebo ze zařízení ukrást citlivé informace pro jejich pozdější zneužití.

1.1 Základní pojmy

Malware je obecný termín pro škodlivé programy, které se řadí do jednotlivých kategorií. Tyto kategorie bývají v odborných literaturách nazývány také jako tzv. rodiny. Rodiny udávají základní charakteristiku malwaru. To znamená, že při vytvoření nového škodlivého programu, nám právě jeho rodina určuje, jak se bude daný program chovat. Při jeho tvorbě vycházíme zpravidla z modifikace některého ze známých kódů. Mezi nejznámější rodiny malwaru patří viry, červi, trojské koně, adware, spyware, rootkity, exploity atd.

Vir

Program, který po napadení systému se vněm dokáže dále šířit obdobně jako biologický virus. Tohoto procesu dosahuje tím, že se vkládá do jiných spustitelných souborů nebo dokumentů.

Červ

Jedná se o zvláštní druh počítačového viru. Šíří se prostřednictvím infikovaných souborů nebo paketů počítačové sítě. V případě napadení systému dokáže odeslat své kopie na další zařízení a ty infikovat. Rozdíl mezi virem a červem je velmi úzký, a to zpravidla ten, že červ se dokáže šířit sám bez účasti hostitele.

Trojský kůň

Název škodlivého programu pochází z antické doby z příběhu o dobytí Tróje. Jedná se o program, který se na první pohled tváří jako užitečný (např. hra, spořič obrazovky nebo jiný užitečný nástroj). Poté co si jej uživatel nainstaluje do svého zařízení v domnění, že se nejedná o škodlivý program, začne provádět jinou činnost, než ke které má být určen.

Činnost původní však může dělat bez jakýchkoliv problémů. Rozdíl mezi počítačovým virem a trojským koněm je ten, že trojský kůň se sám nedokáže šířit na další zařízení.

Adware

Jde o programy, které na infikovaném zařízení uživateli znepříjemňují práci nějakou reklamní činností. To může být v podobě vyskakujících pop-up oken nebo běžných reklamních bannerů. Ve většině případů jsou doprovázeny některými freeware[1] programy.

Spyware

Jedná se o programy, které bez vědomí uživatele odesílají z napadeného zařízení citlivé informace. Může se jednat např. o seznam oblíbených stránek nebo historii internetového prohlížeče, seznam kontaktů, hesla nebo informace o poloze z GPS (Global Position System) přijímače. K tomuto procesu využívají především internetového spojení, což může mít za následek jak jeho zpomalení, tak v případně mobilních zařízení rychlé překročení datového limitu FUP[2] (Fair Use Policy).

Rootkit

Rootkit je nenápadný typ nástroje, který má za úkol skrýt existenci určitých programů nebo procesů v systému tak, aby jiný škodlivý program nebyl běžnými systémovými nástroji nebo bezpečnostními aplikacemi odhalitelný. Toho lze dosáhnout například skrýváním adresářů vybraných programů nebo položek v registru. Existence rootkitu nemusí hned znamenat přímé nebezpečí, ale bývá zpravidla prvním krokem pro napadení systému.

Exploit

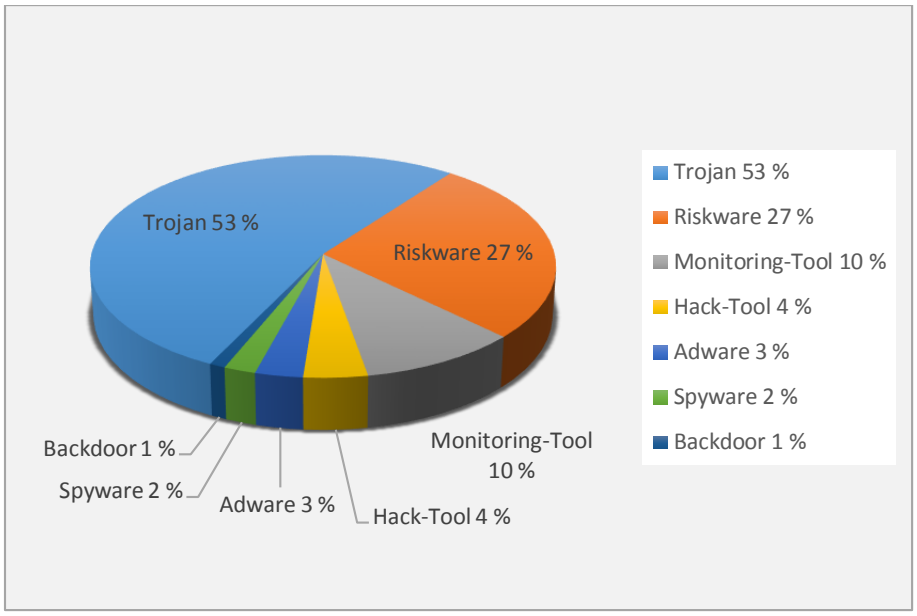
Jedná se o speciální program nebo sekvenci příkazů, který využívá programátorské chyby v systému za účelem jeho ovládnutí nebo instalaci jiného škodlivého programu. Tyto bezpečnostní chyby bývají zpravidla odstraněny aktualizacemi v systému.

---

[1] Freeware je označení pro softwarové programy, které jsou distribuovány zpravidla bezplatně. Pravidla používání bývají zakotvena v licenční smlouvě. Ve většině případů je takový program zakázáno dále šířit nebo upravovat bez vědomí autora.

[2] FUP je způsob sdílení internetového připojení mezi více uživateli. Ti jsou zpravidla po vyčerpání svého datového limitu za určité časové období omezeni tím, že je jim do konce vyúčtovacího období dočasně snížena rychlost připojení.

Na následujícím obrázku je zobrazeno zastoupení mobilního malwaru podle jednotlivých kategorií ve čtvrtém čtvrtletí roku 2012.



Obrázek 1 Zastoupení malwaru podle jednotlivých kategorií [3]

1.2 Historie

Vývoj škodlivých kódů pro mobilní zařízení roste mnohem rychlejším tempem než u klasických počítačových systémů. Přispívá k tomu především skutečnost, že dnešní smartphony jsou přímo propojeny s platebním systémem ať už formou internetového bankovnictví nebo v jednodušší formě s možností zasílání prémiových SMS (Short Message Service) zpráv. Právě tyto placené zprávy jsou pro útočníky velmi lákavé.

Historie mobilního malwaru se počítá již od roku 2000. Jedná se však o první kódy, které měly teprve připravit půdu pro příchod dalších mnohem propracovanějších kódů. Od roku 2004 se objevilo více než 30 různých rodin mobilního malwaru, přičemž modifikace virů známých rodin se vyšplhala na více než několik set. Tento úspěch je dosažen díky více jak 30 letým znalostem z programování klasických počítačových virů [1].

Mezi nejznámější představitele historie mobilního malwaru patří:

- Cabir
- Skulls
- CommWarrior
- Trojan.Redbrowser.A

Cabir

Jeden z prvních škodlivých programů vytvořených pro mobilní zařízení se nazývá Cabir. Tento červ vznikl v roce 2004 a byl určen pro napadení mobilních zařízení s operačním systémem Symbian. Konkrétně se jednalo o telefonní přístroje Nokia 60. Na tomto zařízení se šířil prostřednictvím rozhraní bluetooth a to pomocí souboru SIS[3] (Symbian Installation File). Uživatel musel v takovém případě vždy potvrdit přijetí souboru, aby se zařízení mohlo infikovat. Jakmile bylo provedeno úspěšné nakažení zařízení, dokázal červ sám bez vědomí uživatele vyhledat další zařízení, na které se může sám odeslat. V tomto případě musí napadený uživatel opět potvrdit přijetí souboru [1].

Skulls

Skulls neboli lebky je další škodlivý program tentokrát z rodiny trojských koní. Byl vytvořen v listopadu roku 2004 a byl určen opět pro zařízení s operačním systémem Symbian. Program je distribuován prostřednictvím e-mailu, webových stránek nebo technologie P2P[4] (Peer To Peer). Instalace škodlivého kódu musí být i v tomto případě potvrzena uživatelem. V případě nakažení se na zařízení nahradí všechny ikony obrázky lebek s překříženými hnáty. Na zařízení bude dále možné pouze přijímat a provádět hovory. Ostatní aplikace již fungovat nebudou [1].

CommWarrior

Program byl poprvé identifikován v roce 2005 a byl zajímavý především možností šíření jak přes rozhraní bluetooth, tak prostřednictvím MMS (Multimedia Messaging Service) zpráv. Byl také určen pro zařízení se systémem Symbian a šířil se v souborech s příponou SIS. Tento červ se dokázal sám bez vědomí uživatele rozesílat na kontakty z adresáře zařízení prostřednictvím MMS zpráv. Uživatel si přitom ničeho nemusel všimnout a na přítomnost červa ho upozornil až větší poplatek za tyto zprávy [1].

Trojan.Redbrowser.A

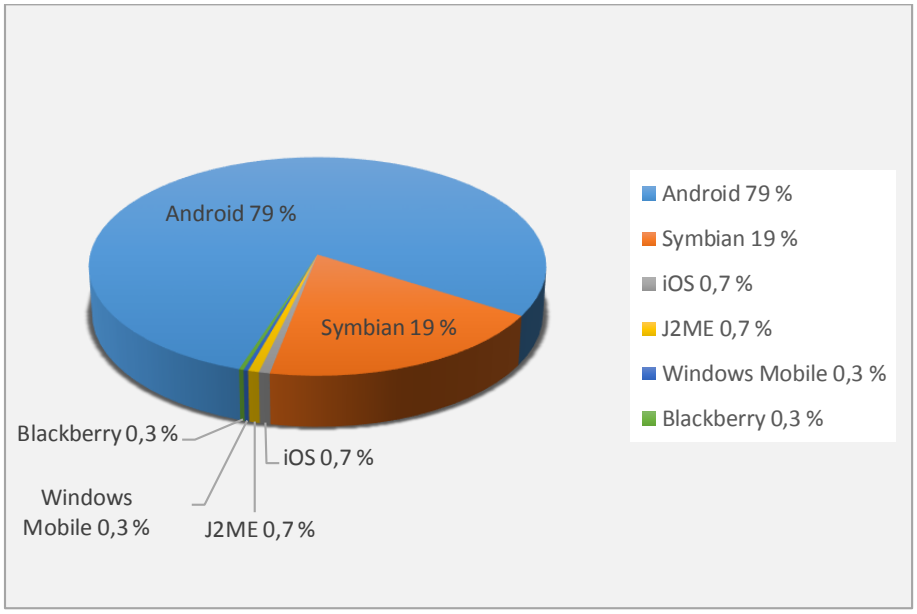
---

[3] Soubory s koncovkou SIS jsou instalační soubory operačního systému Symbian, který je nejčastěji používán u starších telefonů Nokia.

[4] Označení peer to peer (klient-klient) slouží pro označení typu počítačové sítě, kdy spolu jednotlivý uživatelé komunikují jako rovný s rovným bez přítomnosti serveru.

Tento trojský kůň byl objeven v únoru roku 2006 a byl prvním objeveným programem, který napadal mobilní zařízení běžících na J2ME[5] (Java 2 Micro Edition) platformě. Program se navenek tváří jako bezpečná aplikace pro přístup k internetu. Jeho skrytou funkcí je však odesílat SMS zprávy na prémiová čísla. Jelikož každou takovou transakci musí povolit uživatel, je míra nebezpečí minimální [2].

V historii se mobilní malware zaměřoval především na zařízení s platformou Symbian, která byla v té době velmi používaná. Od roku 2010, kdy se do popředí dostává operační systém Android, prochází i mobilní malware velkými změnami a zaměřuje se právě na tento operační systém. V současné době je pro systém Android registrována celá řada škodlivých kódů. Na následujícím obrázku je zobrazeno rozšíření mobilního malwaru podle jednotlivých platforem v roce 2012.

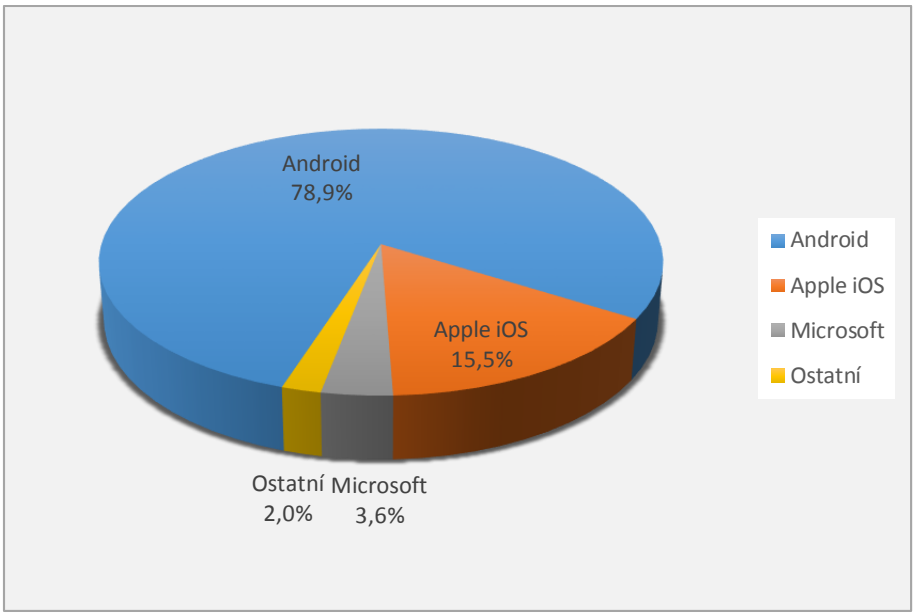


Obrázek 2 Rozšíření malwaru podle jednotlivých platforem [3]

[5] J2ME je jednou ze základních platforem Javy určených pro mobilní zařízení. Platforma nabízí robustní a flexibilní prostředí pro běh aplikací na zařízeních, kde je omezené množství paměti nebo malá energetická kapacita.

2 OPERAČNÍ SYSTÉM ANDROID

Operační systém Android je nejrychleji se rozšiřujícím operačním systémem současnosti vytvořeným především pro mobilní zařízení. I když je z velké části systém využíván především v chytrých telefonech, lze jej nalézt také v tabletech, PDA zařízeních, televizích nebo navigacích. V současné době je vyvíjen společenstvím OHA[6] (Open Handset Alliance) a jedná se o otevřenou platformu založenou na jádru Linux[7]. Právě otevřená platforma dává vývojářům široké možnosti v co možná nejefektivnějším využitím tohoto systému. Na obrázku 3 je zobrazen podíl jednotlivých platforem na trhu mobilních telefonů v roce 2013.



Obrázek 3 Podíl smartphonů na trhu v roce 2013 dle platforem [4]

[6] Open Handset Alliance je konsorcium společností zabývajících se mobilními technologiemi, výrobou mobilních telefonů a telekomunikačních operátorů za účelem urychlení inovace mobilních zařízení a poskytnutí kvalitnějšího prostředí a služeb jak pro uživatele, tak pro vývojáře mobilních platforem. Aliance vznikla 5. 7. 2007 a v současné době má více jak 80 členů.

[7] Linux je označení pro volně šiřitelnou platformu operačního systému vycházejícího z principu unixových systémů. Jeho různé distribuce je na základě volné licence možné dále šířit i upravovat.

2.1 Historie

Významnou postavou v historii Androidu je Andy Rubin, který v říjnu roku 2003 založil společnost Android Inc. Tu v roce 2005 odkoupila společnost Google Inc. a udělala z ní svoji dceřinou společnost. V roce 2007 Google získal několik patentů v oblasti mobilních technologií a začaly se šířit spekulace, že společnost chce vstoupit na trh chytrých mobilních telefonů. Platforma Android byla ohlášena v listopadu roku 2007 v den, kdy byla založena aliance OHA, která zahrnovala přední světové výrobce, jako jsou Google, Samsung, HTC, Intel, LG, NVIDIA, Sony, Qualcomm, T-Mobile a další [5].

Přehled jednotlivých verzí OS Android [6] [7]:

Android 1.0 (2008)

První verze OS Android byla představena v září roku 2008 a byla zpřístupněna především vývojářům, aby se mohli se systémem seznamovat a postupně připravovat nové aplikace. Ten samý rok byl Android uvolněn jako Open source platforma[8]. Prvním telefonem s tímto operačním systémem byl model HTC Dream 100.

Android 1.1 (2008)

Tato verze měla především za úkol opravit chyby verze předchozí a zpřístupnit možnost využití služby Android Market i pro placené aplikace.

Android 1.5 Cupcake (2009)

Hlavním přínosem verze byla možnost nahrávat a sledovat videa prostřednictvím režimu videokamery a nahrávat videa a fotografie na stránky YouTube a Picaso přímo z telefonu. Dále byla s příchodem verze zpřístupněna možnost umístit na obrazovku Widgety[9] a provádět animace při přechodu mezi obrazovkami.

Android 1.6 Donut (2009)

---

[8] Open Source Platformou neboli Open Source Softwarem je počítačový software s otevřeným zdrojovým kódem. V takovém případě je uživateli umožněno za jistých podmínek software bezplatně využívat i upravovat.

[9]Widgety jsou v případě mobilních zařízení malé ovládací prvky, pomocí kterých lze zobrazit doplňující informace na obrazovce zařízení (např. počasí nebo čas), nebo pomocí kterých lze spustit příslušný program.

Tato verze poskytovala vylepšení služby Android Market a integraci nového prostředí fotoaparátu, kamery a galerie. Dále bylo ve verzi aktualizované rozhraní vyhledávání hlasem a byla zde zpřístupněna možnost rozšířeného vyhledávání přímo z hlavní obrazovky.

Android 2.0, 2.1 Eclair (2009)

Verze Enclair poskytovala vylepšené uživatelské prostředí, optimalizaci výkonu hardwaru, možnost použití animovaných tapet na domovské obrazovce, podporu bluetooth 2.1, digitální zoom fotoaparátu, podporu pro Microsoft Exchange a vylepšené mapy Google.

Android 2.2 Froyo (2010)

S příchodem verze Froyo bylo zpřístupněno sdílení internetu přes rozhraní Wifi nebo USB (Universal Serial Bus) a byla povolena možnost instalovat plugin Adobe Flash 10.1, který slouží pro zobrazení vektorové grafiky a zpřístupnění interaktivního multimediálního obsahu webových stránek.

Android 2.3 – 2.3.7 Gingerbread (2010)

Verze zpřístupňuje podporu videoformátu WebM[10], podporu standardu NFC[11] (Near Field Communication), vylepšení funkcí pro sociální sítě, podporu více kamer a podporu nových senzorů jako je gyroskop.

Android 3.0 – 3.2 Honeycomb (2011)

Verze byla zaměřena na podporu tabletů. Kromě vylepšení uživatelského rozhraní byla ve verzi upravena podpora multitaskingu[12], podpora vícejádrových procesorů a podpora hardwarové akcelerace.

Android 4.0 – 4.0.4 Ice Cream Sandwich (2011)

---

[10] WebM je audio-video kontejner založený na formátu Matroska, který umožňuje otevřenou kompresi videa pro použití v jazyce HTML5.

[11] NFC je technologie sloužící jako standard pro bezdrátovou komunikaci mezi elektronickými zařízeními na krátkou vzdálenost. V současné době je preferovaná především u bezkontaktních finančních transakcí.

[12] Multitasking je schopnost operačního systému umožňující běh více procesů současně. V tomto případě uživatel může mezi procesy nebo aplikacemi přepínat, aniž by je musel zcela ukončit.

Tato verze přináší mimo jiná vylepšení především jednotné rozhraní pro telefony a tablety, možnost odemčení telefonu obličejem prostřednictvím přední kamery a vylepšenou funkci Android Beam, prostřednictvím které je možné přenášet obsah mezi dvěma telefony pouhým přiblížením.

Android 4.1 – 4.3 Jelly Bean (2012)

Verze přináší rychlejší a hladší orientaci v prostředí systému. Od verze 4.3 je možné využívat více účtů a tyto účty také spravovat. Dále verze přináší podporu OpenGL[13] (Open Graphics Library), díky které je zpřístupněna funkce Project Butter, která zajišťuje rychlejší a plynulejší vykreslování obrazu. Z bezpečnostního hlediska je značným přínosem nová funkce pro kontrolu aplikací Verify apps, která byla zpřístupněna ve verzi Android 4.2.

Android 4.4 KitKat (2013)

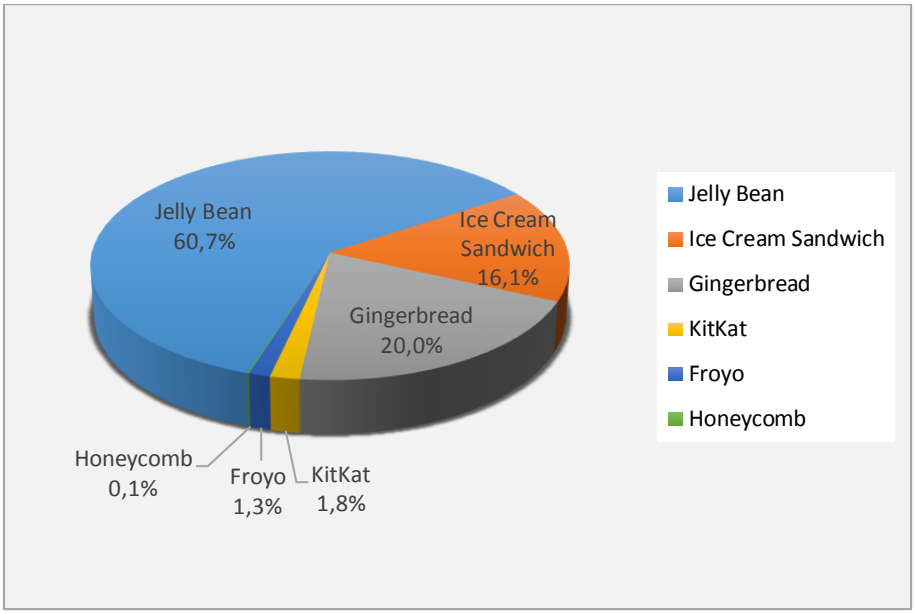
Zatím nejnovější verze operačního systému, která s sebou přináší odstranění nástroje na ochranu soukromí. Jedná se o nástroj, který umožňuje uživatelům zablokovat shromažďování osobních informací některými aplikacemi. Dle vyjádření Googlu byla tato funkce zavedena do systému omylem ve verzi 4.3, a proto je z této verze odstraněna. Další novinkou verze je možnost platit mobilním telefonem v obchodě prostřednictvím bezkontaktní technologie HCE[14] (Host Card Emulation).

Od roku 2008 vyšlo mnoho verzí tohoto operačního systému, kde s každou novou verzí jsou odstraňovány chyby verzí předchozích a jsou doplňovány další užitečné funkce. Na následujícím obrázku je zobrazeno zastoupení jednotlivých verzí dle kódového označení. Data jsou aktualizována k únoru 2014. Verze, které mají distribuci menší jak 0,1 %, zde zobrazeny nejsou.

---

[13] OpenGL představuje standardní multiplatformní rozhraní API využívající se pro tvorbu aplikací počítačové grafiky. Je využíváno pro komunikaci s grafickým procesorem (GPU - Graphic Processing Unit) čímž je dosaženo hardwarově akcelerovaného vykreslování grafiky.

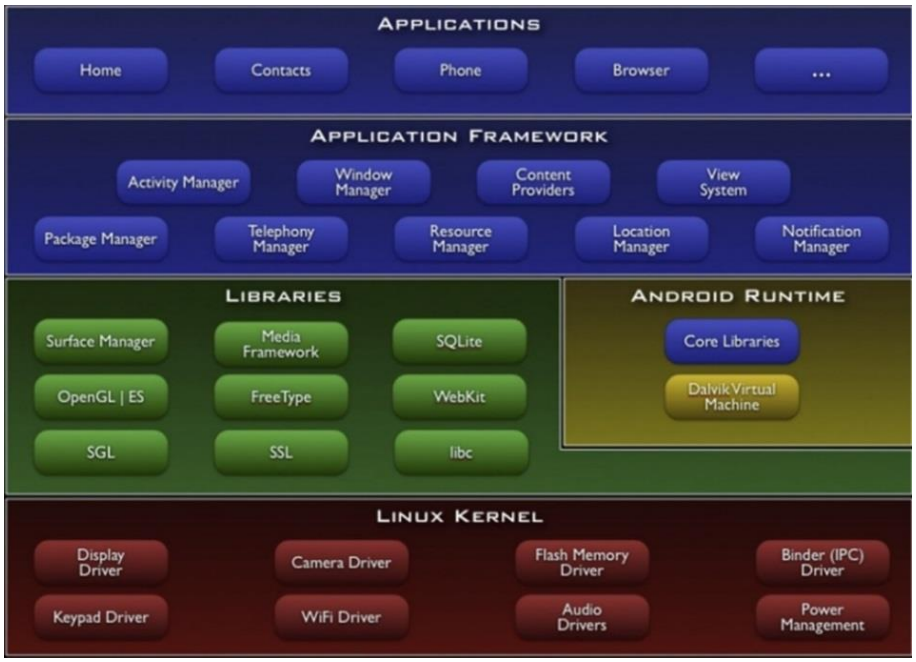
[14] HCE slouží jako bezpečnostní prvek při využívání bezdrátové transakční technologie NFC. Jedná se o softwarovou emulaci čipové karty.



Obrázek 4 Zastoupení jednotlivých verzí OS Android [7]

2.2 Základní architektura OS Android

OS Android je otevřenou platformou, kdy jeho architektura je postavena z několika vrstev. Základních vrstev je pět, přičemž každá vrstva nižší úrovně poskytuje služby vrstvě úrovně vyšší. Tyto vrstvy jsou zobrazeny na obrázku 5.



Obrázek 5 Model vrstev OS Android [5]

2.2.1 Linux Kernel (Jádro Linux)

Jádro je nejnižší abstraktní vrstvou, která zprostředkovává komunikaci mezi používaným hardwarem a zbytkem softwaru. Systém vznikl původně na jádru Linux verze 2.6 a v současné době ve verzi OS Android 4.x je postaven na jádře Linux verze 3.x. Linux poskytuje bezpečné a jednoduché uživatelské rozhraní s nutností nastavení oprávnění. Tato oprávnění jsou velmi důležitá z hlediska bezpečnosti systému, jelikož brání čtení informací z jiné aplikace, než pro kterou jsou určeny. Mezi hlavní funkce jádra patří komunikace s hardwarem, pro kterou je opatřen ovladači, dále zajišťuje správu procesů, správu paměti a síťové spojení [8].

2.2.2 Libraries (Knihovny)

Android obsahuje C a C++ sadu nativních knihoven. Funkce systému využívající různé komponenty hardwaru jsou uloženy v této vrstvě a vývojáři k nim přistupují prostřednictvím těchto knihoven přes aplikační rámec (Application Framework) [8][9].

2.2.3 Android Runtime

Tato vrstva se nachází na stejné úrovni s vrstvou knihoven a skládá se ze dvou základních částí. První je virtuální stroj Dalvik VM (Virtual Machine), na kterém jsou spuštěny všechny běžící aplikace. Dalvik je upravený JVM[15] (Java Virtual Machine) lišící se do jisté míry architekturou a příponami souborů, které spouští. Dalvik spouští soubory s příponou (dex) na rozdíl od JVM, který spouští soubory s příponou (class). Druhou částí vrstvy jsou systémové knihovny Core Libraries, jenž jsou základní knihovny programovacího jazyka Java[16], které slouží pro všeobecné programovací rozhraní API[17] (Application Programming interface) [8].

---

[15] JVM je virtuální stroj, který převádí binární kód jazyka Java (tzv. Java bytecode) do strojového kódu, který je následně spustitelný na určité mikroprocesorové architektuře nebo v určitém operačním systému. Důvodem jeho použití je možnost spouštět programy a aplikace navrženy pro jazyk Java i na jiných platformách, aniž by musely být překládány nebo přepsány zvlášť pro jednotlivé platformy.

[16] Programovací jazyk Java je používán pro programy, které pracují na různých systémech. Jedná se například o systémy pro zařízení, jako jsou telefony, tablety, čipové karty nebo systémy pro desktopové počítače.

2.2.4 Application Framework (Aplikační rámec)

Aplikační rámec poskytuje bohatou sadu modulů určených pro vývojáře. Každý z těchto modulů nabízí určitou skupinu služeb pro správu systému. Mezi nejdůležitější moduly této vrstvy patří [8]:

- Aktivity Manager – jedná se o modul správy aktivit, který řídí životní cyklus každé aplikace.
- Package Manager – je modul správy balíčků, který je zodpovědný za instalaci aplikací v systému a uchování informací o již nainstalovaných aplikacích.
- Resource Manager – modul zodpovědný za poskytnutí přístupu ke zdrojům jednotlivých souborů dané aplikace. Jedná se např. o soubory grafiky případně jiné řetězce kódů vyžadované každou aplikací.
- Notification Manager – modul umožňující aplikacím v oznamovací liště zobrazovat upozornění na vybrané události v systému. Jedná se např. o upozornění na přijetí nové zprávy nebo e-mailu.

2.2.5 Applications (Aplikace)

Aplikační vrstva je nejvyšší vrstvou architektury. Vrstva nabízí aplikace, které jsou buď součástí systému, nebo jsou do systému dodatečně nainstalovány uživatelem. Mezi základní aplikace, jenž jsou součástí systému patří kalendář, správce kontaktů, webový prohlížeč nebo SMS klient. Aplikace jsou napsány v kódu Java s příponou (class) a distribuovány prostřednictvím instalačních balíčků s příponou APK (Android Application Package) [8].

2.3 Bezpečnostní prvky systému

Bezpečnost operačních systémů se dá rozdělit do čtyř základních skupin. Tyto skupiny obsahují jak bezpečnostní prvky na úrovni systému a jádra, tak bezpečnostní prvky na úrovni aplikací.

---

[17] Rozhraní API je soubor funkcí, protokolů a proměnných, které se nacházejí v knihovnách a určují, jakým způsobem jsou funkce knihovny volány ze zdrojového kódu programu. API rozhraní je využíváno vývojáři pro programování aplikací.

Základní skupiny bezpečnosti systému Android jsou:

- System Security (Bezpečnost systému)
- Encryption a Data Protection (Šifrování a ochrana dat)
- Apps Security (Bezpečnost aplikací)
- Update (Aktualizace)

2.3.1 System Security (Bezpečnost systému)

Základní ochrana systému je jednou z nejdůležitějších ochran, co se bezpečnosti týče. Jedná se o celkovou ochranu integrity systému, do které patří tyto bezpečnostní prvky:

- Password (Heslo)
- System Partition a Safe Mode (Systémový oddíl a nouzový režim)
- Filesystem Permissions (Oprávnění souborového systému)
- Memory Management Security (Bezpečnost správy paměti)

Password (Heslo)

Hlavní heslo, označováno také jako uživatelské heslo, je zásadním bezpečnostním prvkem systému, které je využíváno i k dalším bezpečnostním funkcím. Toto heslo by rozhodně nemělo být uživatelem opomíjeno. V případě, kdy je heslo aktivováno, je uživatel nucen při každém vstupu do systému toto heslo zadat.

System Partition a Safe Mode (Systémový oddíl a nouzový režim)

System Partition je systémový oddíl, který obsahuje soubory potřebné pro načtení systému. Do systémového oddílu spadá celá výše popsaná architektura. Tento oddíl je nastaven pouze pro čtení a běžný uživatel nemá oprávnění v něm zapisovat. Toto privilegium je povoleno pouze uživateli s oprávněním správce [10].

Safe Mode je nouzový režim systému. Tento režim může být vyvolán uživatelem v případě, kdy uživatel nemůže ze systému odinstalovat škodlivý software. V případě vyvolání nouzového režimu se přístroj spustí v základním nastavení pouze s předinstalovanými aplikacemi výrobce. Tímto způsobem může uživatel identifikovat nebo odstranit nežádoucí software, aniž by se ten v zařízení spustil [2].

Filesystem Permissions (Oprávnění souborového systému)

Ve výchozím nastavení každá aplikace běží v jádru systému pod vlastním jedinečným číslem UID (Unique Identifier). Aplikace je při instalaci nainstalována do adresáře, kde má

přístup k souborům pouze ona sama. To znamená, že pouze oprávněná aplikace může sdílet data v tomto adresáři, který je opatřen příslušným UID. Tím je zabráněno, aby kterákoliv aplikace přistupovala k datům aplikace jiné. To je možné pouze prostřednictvím zvláštních oprávnění v souborovém systému. Tato oprávnění musí předem nastavit uživatel již při tvorbě aplikace [9].

V případě externích úložišť tato data nepodléhají souborovému systému a přístup k nim může mít jakýkoliv uživatel. Stačí pouze předělat kartu do kteréhokoliv jiného zařízení. Proto je vhodné data na externích úložištích šifrovat, což umožňují vyšší verze operačního systému.

Memory Management Security (Bezpečnost správy paměti)

Jako další bezpečnostní prvek slouží ochrana správy paměti. Zde byla významným přínosem nová technologie ASLR (Address Space Layout Randomization), která byla poprvé použita ve verzi Android 4.0. Technologie měla za úkol znemožnit některé typy útoků především ve formě exploitů. Principem bylo zajistit, aby data nahraná do paměti při startu operačního systému byla náhodně rozmístěna, a tudíž nebylo možné zjistit, kde se právě nachází [10].

Dalším přínos v oblasti správy paměti přinesla verze Android 4.1. Jedná se o podporu PIE (Position Independent Executable), která umožňuje vykonat strojový kód nezávisle na tom, na jaké adrese se v operační paměti nachází, a to v každém okamžiku, kdy je tento kód vykonán. Tato technologie slouží jako doplněk ASLR a musí jí hardwarově podporovat procesor. [10].

2.3.2 Encryption a Data Protection (Šifrování a ochrana dat)

Android poskytuje sadu kryptografických metod dostupných prostřednictvím programovacího rozhraní API. V tomto případě jsou využívány běžné algoritmy jako je AES[18] (Advanced Encryption Standard), RSA[19] (Rivest Shamir Adleman), DSA[20] (Digital

[18] AES je standard pro symetrickou blokovou šifru vytvořený v roce 1998. Pro šifrování i dešifrování využívá stejný klíč s pevnou délkou bloku.

Signature Algorithm), SHA[21] (Secure Hash Algorithm) popřípadě SSL[22] (Secure Sockets Layer) nebo HTTPS[23] (Hypertext Protocol Secure). Tyto algoritmy systém využívá pro šifrování souborového systému (Filesystem Encryption) nebo pro vytvoření šifrované komunikace v některých aplikacích [10].

Z hlediska možností zabezpečení síťového provozu umožňuje systém šifrovat komunikaci přes rozhraní Wi-Fi nebo Bluetooth za pomocí standardních protokolů. Další výhodou vyšších verzí systému je možnost vytvářet privátní sítě VPN[24] (Virtual Private Network).

Filesystem Encryption (Šifrování souborového systému)

Šifrování dat souborového systému bylo zpřístupněno od verze Android 3.0. K tomuto účelu Android využívá symetrických šifer, které mají pevně stanovenou délku klíče pro šifrování i dešifrování. Šifrování v tomto případě probíhá na úrovni jádra. Tyto nástroje jsou implementovány v kryptografickém rozhraní API. Pro šifrování je využíván algoritmus AES s délkou klíče 128 bitů. Hlavní klíč je chráněn pomocí algoritmu AES se 128 bitovým klíčem, který je odvozený od uživatelského hesla, které uživatel zadává pro přihlášení do systému [10] [11].

Pro zajištění odolnosti hesla vůči systematickým útokům hrubou silou případně slovníkovým útokům je použita tzv. Kryptografická sůl, pomocí které se několik

---

[19] RSA je prvním algoritmem používaným v asymetrické kryptografii, který je vhodný jak pro šifrování, tak pro podepisování. Jedná se o šifru s veřejným klíčem, která se při dostatečné délce klíče považuje za bezpečnou dodnes. Název je odvozen podle jmen svých autorů (Rivest, Shamir, Adleman).

[20] DSA je algoritmus pro digitální podpis, který byl americkou vládou prohlášen za standard. Je jedním z hlavních nástrojů pro identifikaci a autentizaci osob v prostředí internetu.

[21] SHA je hashovací funkce, která ze vstupních dat libovolné délky vytváří otisk pevné délky. Používá se např. pro kontrolu integrity souborů nebo pro ukládání hesel.

[22] SSL je protokol využívající se nejčastěji pro bezpečnou šifrovanou komunikaci prostřednictvím webového prohlížeče na principu asymetrického šifrování.

[23] HTTPS je vylepšený protokol HTTP, který využívá asymetrické kryptografie a poskytuje zabezpečené spojení mezi webovým prohlížečem a serverem.

[24] VPN slouží pro spojení dvou nebo více privátních sítí prostřednictvím sítě veřejné jako je např. internet. Toto spojení je zprostředkováno přes server VPN a z pohledu uživatele se tyto sítě tváří jako jedna privátní.

náhodným bitů doplní dalšími bity, a to při každém opakovaném vstupu do hashovací funkce SHA-1. To zaručí, že stejné heslo bude mít pokaždé jiný zakódovaný tvar. K tomu je využito PBKDF2[25] (Password Based Key Derivation Function) algoritmu [10].

V případě použití internetové asynchronní kryptografie využívá Android šifry s veřejným klíčem RSA za použití algoritmu digitálního podpisu DSA. To nám zaručuje důvěrnost informací a ochranu proti neautorizované modifikaci nežádoucí osobou.

2.3.3 Apps Security (Bezpečnost aplikací)

Android dovoluje osobám třetích stran vyvíjet a distribuovat pro tento systém aplikace. Tyto aplikace je možné zpřístupnit ostatním uživatelům buď prostřednictvím služby Google Play[26], kterou provozuje společnost Google, anebo prostřednictvím svých webových stránek. Každá aplikace však musí projít základním bezpečnostním procesem a musí obsahovat určitá bezpečnostní opatření. Mezi zásadní bezpečnostní prvky týkajících se ochrany aplikací patří:

- Sandboxing (Izolovaný prostor)
- Application Signing a Verification (Podpis aplikací a ověření)
- Access Permissions API (Oprávnění přístupu API)
- Google Bouncer

Sandboxing (Izolovaný prostor)

Každá aplikace v systému běží v tzv. Sandbox módu, což znamená, že běží v izolovaném prostředí. V některých literaturách se to nazývá také jako tzv. aplikační karanténa. O tuto karanténu se systém postará tak, že každé aplikaci přiřadí jedinečné číslo UID (Unique Identifier) a spustí ji v samostatném procesu. Jádro systému (kernel) pomocí čísla UID rozliší uživatelská oprávnění přiřazená zvlášť každé aplikaci. To zajistí, že aplikace nemohou mezi sebou komunikovat a mají také omezený přístup k operačnímu systému. O

[25] PBKDF2 je algoritmus aplikující pseudonáhodnou funkci na vstupní heslo, kdy spolu s tzv. kryptografickou solí je opakujícím se procesem vytvořen nový kryptografický klíč.

[26] Služba Google Play vznikla 6. 3. 2012 sloučením služby Google Music a Android Market.

případnou komunikaci mezi aplikacemi běžících na různých procesech se stará IPC[27] (Inter-Process Comunication), který je součástí nejnižší vrstvy architektury. Tato komunikace však musí být vývojářem předem nastavena tak, že se každé aplikaci přiřadí určitá oprávnění a ty pak prostřednictvím IPC mohou mezi sebou vzájemně komunikovat. Tento způsob se liší od jiných, především desktopových operačních systémů, kde několik aplikací může běžet se stejnými oprávněními [10].

Application Signing a Verification (Podpis aplikací a ověření)

Každá aplikace nainstalovaná v systému Android vyžaduje, aby byla autorem digitálně podepsána. To umožňuje identifikovat autora každé aplikace, vytvářet vztahy důvěryhodnosti mezi aplikacemi a provádět jejich aktualizace bez nutnosti nastavování složitých oprávnění. Aplikace jsou podepsány pomocí certifikátů, jejichž soukromý klíč si v tomto případě generují samotní uživatelé. Certifikát přitom nemusí být podepsán certifikační autoritou. To je asi největší kámen úrazu, jelikož není zaručena věrohodnost těchto autorů. Ti mohou již vytvořenou aplikaci modifikovat, vložit do ní malware a následně podepsat svým novým certifikátem. Potom jim nic nebrání, aby škodlivou aplikaci zveřejnili na některé ze svých webových stránek a tím zpřístupnili veřejnosti. Proces ověření podpisu přitom funguje tak, že v případě instalace aplikace na zařízení provede modul správy balíčku Package Manager ověření, jestli byl soubor (APK) řádně podepsán a v případě že ano, povolí instalaci. U každé aplikace, která není autorem podepsána, bude její instalace odmítnuta, případně bude zamítnuta její distribuce v Google Play obchodě [10].

Funkce pro kontrolu aplikací je poprvé zpřístupněna ve verzi Android 4.2 a funguje na principu ověření prostřednictvím Google serveru. Zde se ověří, zda je aplikace potencionálním nebezpečím či nikoli. V kladném případě se zobrazí dvě možná hlášení. První hlášení upozorní na detekci škodlivé aplikace a nabídne možnost přerušení nebo pokračování v instalaci. Na uživateli potom záleží, zda aplikaci nainstaluje nebo ne. Druhé hlášení upozorní na detekci nebezpečné aplikace a instalaci automaticky přeruší. Princip je

[27] IPC je zkratka, která se v informatice zavádí pro komunikaci mezi více procesy. V případě operačního systému Android je tato komunikace obstarávána na úrovni jádra.

přitom založen na odeslání a následném ověření zjištěných hodnot o aplikaci, jako je hodnota SHA-1, jméno aplikace, verze aplikace atd.

Access Permissions API (Oprávnění přístupu API)

Bezpečnost systému na aplikační úrovni je založena na oprávněních. Ta jsou nedílnou součástí každé aplikace. Setkávají se s nimi jak uživatelé, tak vývojáři aplikací. Každá aplikace, aby mohla komunikovat s jinou aplikací a aby mohla využívat systémových prostředků, musí mít uživatelem nastavena právě tato oprávnění. K oprávněním je možné přistupovat pomocí chráněného API rozhraní. Existují však API výjimky, které nejsou systémem podporovány (např. přímý přístup k SIM kartě). Mezi prostředky chráněného API patří [10] [11]:

- Funkce fotoaparátu
- Funkce SMS/MMS
- Údaje o poloze GPS
- Telefonní funkce
- Funkce Bluetooth
- Funkce sítí a datového spojení

Při instalaci aplikací se uživateli zobrazí navigační okno se seznamem oprávnění, která aplikace vyžaduje pro svoji instalaci. Uživatel tato oprávnění musí potvrdit všechna bez výjimky. Nestačí vybrat pouze některé z nich. V takovém případě se aplikace nenainstaluje. Seznam veškerých oprávnění, která mohou aplikace využívat, je obsažen v souboru *Manifest* nacházejícím se v instalačním balíčku (APK). Do tohoto souboru je musí nadefinovat vývojář již při vytváření každé aplikace. Aplikaci, která se během svého provozu pokusí využít funkci, ke které nemá přidělená oprávnění, bude přístup odepřen. Podrobnější výklad zabývající se problematikou oprávnění je obsažen v praktické části práce [10].

Google Bouncer

Bouncer je testovací skener, který v současné době společnost Google používá pro automatické testování svých aplikací přímo v Google Play obchodě. Úkolem skeneru je ve virtuálním prostředí testovat aplikace, zda vykazují přítomnost jakékoliv anomálie, která by se mohla jevit jako škodlivý program. V takovém případě aplikaci z obchodu vyřadí.

Dalším úkolem je testovat nové vývojářské účty a zabránit tomu, aby vývojáři, kteří již v minulosti škodlivé aplikace vytvářeli, nemohli v této činnosti pokračovat [12].

2.3.4 Updates (Aktualizace)

Aktualizace jsou z hlediska bezpečnosti velmi důležité, protože opravují jak chyby aplikací třetích stran, tak chyby obsažené přímo v operačním systému. Samotný systém lze aktualizovat dvěma možnými způsoby. První způsob je prostřednictvím programu přímo v počítači. Tento program slouží zároveň pro správu zařízení a každý výrobce používá svůj vlastní. V takovém případě se z centrálního umístění výrobce stáhne aktualizační soubor (ZIP), který se do telefonu nainstaluje prostřednictvím počítače. Před tímto procesem je nejprve systémem ověřena integrita a autentičnost instalačního balíčku a až poté je systém aktualizován. Druhým způsobem je aktualizace prostřednictvím bezdrátového připojení OTA[28] (Over The Air) [10].

V případech, že je v systému objeveno jakékoliv bezpečnostní riziko, může jej vývojář nebo uživatel nahlásit bezpečnostnímu týmu Android např. prostřednictvím e-mailu. Ten se po takovém upozornění postará o opravu problému a vydá aktualizaci. Tato aktualizace je poskytnuta jednotlivým výrobcům zařízení, kteří se musí postarat, aby se dostala až ke koncovým uživatelům. V některých případech může nastat problém, a to pokud některý výrobce reaguje se značným zpožděním a aktualizaci vydá, až když je chyba zveřejněna. Dalším problémem může být skutečnost, že někteří výrobci vůbec neumožnují přechod na vyšší verzi systému.

[28] OTA je bezdrátová technologie používající se pro dodání nového softwaru nebo dat pro mobilní zařízení jako jsou telefony nebo tablety. Tímto způsobem lze aktualizovat jak obsah jednotlivých aplikací, tak firmware zařízení.

3 OPERAČNÍ SYSTÉM IOS

Operační systém iOS je vyvíjen společností Apple Inc. a na rozdíl od systému Android je uzavřenou platformou. To znamená, že jeho zdrojový kód není poskytován ostatním výrobcům mobilních zařízení. Z tohoto důvodu je také omezena možnost pořízení si jiného zařízení s tímto operačním systémem než takového, který vyrábí právě společnost Apple. Těchto zařízení společnost neposkytuje mnoho a zákazník je nucen si vybrat mezi chytrými telefony iPhone, MP3 přehrávači iPod nebo tablety iPad. Stejně jako u systému Android nekomunikuje hardware s aplikacemi přímo, ale přes operační systém prostřednictvím předem definovaného systémového rozhraní. I v tomto případě je hlavní funkcí operačního systému správa hardwaru a poskytování technologií aplikacím.

3.1 Historie

První verzi systému představil zakladatel společnosti Apple Steve Jobs dne 9. 2. 2007 společně s novým mobilním telefonem iPhone. Systém byl představen pod názvem iPhone OS 1.0 a byl určen pouze pro tyto telefony. Další verze přinášely spousty vylepšení jak ve formě uživatelského komfortu, tak v otázce bezpečnosti.

Základní verze systému iOS jsou [13]:

iPhone OS 1.0 (2007)

První verze systému, která byla učena pouze pro telefony iPhone. V té době se jednalo o revoluční systém pro mobilní telefony. Verze však nepodporovala multitasking, 3G síť ani používání aplikací třetích stran. Jednalo se o zcela uzavřený systém jak před hackery, tak před vývojáři aplikací. Poprvé však byla představena klávesnice zobrazující se přímo na displeji zařízení.

iPhone OS 2 (2008)

Nová verze přišla v červenci 2008 a Apple zde představil službu App Store[29], která umožňovala do zařízení instalovat a používat aplikace třetích stran. K tomu mělo pomoci

[29] App Store je služba společnosti Apple poskytující aplikace pro systémy iOS. Aplikace jsou poskytovány prostřednictvím internetového obchodu a jsou řazeny do různých kategorií podle svého zaměření. Služba je obdobou služby Google Play pro systémy Android.

nové prostředí SDK (Software Development Kit), které mělo pomoci vývojářům k vytváření aplikací a zároveň k jejich testování ve virtuálním prostředí. Dalším vylepšením byla podpora rozhraní Microsoft Exchange, což umožňovalo synchronizaci kontaktů, kalendářů a e-mailů prostřednictvím mobilních zařízení.

iPhone OS 3 (2009)

Třetí verze systému byla zpřístupněna v červnu roku 2009 a podporovala zasílání zpráv MMS. Další podporou byla funkce hlasového ovládání, funkce vyhledávání obsahu a funkce push notifications[30], prostřednictvím které bylo možné zobrazovat upozornění pro aplikace třetích stran. Z hlediska bezpečnosti byla poprvé zpřístupněna služba Find My iPhone, pomocí které uživatel mohl lokalizovat zařízení na dálku prostřednictvím webového prohlížeče. Další výhodou této řady byla podpora pro zařízení iPad, kterou přinesla verze iPhone OS 3.2.

iOS 4 (2010)

Čtvrtá verze systému byla představena v červnu roku 2010 a přišla spolu s novým mobilním telefonem iPhone 4. Hlavní odlišností této verze byla změna názvu z iPhone OS na iOS a podpora nového displeje Retina, který je použit právě v zařízení iPhone 4. Další významnou změnou v systému bylo použití multitaskingu a možnost konverzace prostřednictvím videohovorů Face Time. Z hlediska bezpečnosti byla poprvé ve verzi iOS 4.3 použita technologie ASLR a PIE.

iOS 5 (2011)

V říjnu roku 2011 byla vydána verze iOS 5, která s sebou přinesla spousty novinek a vylepšení. Mezi ty nejvýznamnější patří služba iMessage, prostřednictvím které uživatelé mohli odesílat textové zprávy zdarma přes datovou síť. Služba byla přitom implementovaná do původní aplikace SMS zpráv. Dalším přínosem byla služba hlasové asistentky Siri, pomocí které lze zařízení ovládat bez fyzického kontaktu pouze hlasem. Tato služba však byla přístupná pouze na zařízeních iPhone 4S a vyšší, jelikož byly vybaveny novou generací mikrofonu, který byl pro tuto službu klíčový. Dalším vylepšením

[30] Push notifications je služba, pomocí které může aplikace upozornit uživatele na příchozí událost (např. příchozí e-mail nebo možnost aktualizace).

verze byla podpora synchronizace se softwarem iTunes prostřednictvím sítě Wifi a zpřístupnění služby iCloud[31].

iOS 6 (2012)

Verze iOS 6 byla představena v červnu roku 2012 a mezi její hlavní změny patří výměna aplikace Mapy, patřící pod správu společnosti Google za vlastní mapy Apple. Dalším vylepšením této verze byla integrace sociální sítě Facebook přímo do systému nebo sdílení fotografií na sociálních sítí přímo ze zařízení. V této verzi byla také přidána nová aplikace Passbook sloužící pro centrální umístění elektronických platidel, jako jsou lístky, letenky, vstupenky atd.

iOS 7(2013)

Zatím nejnovější verze systému byla vydána v červnu 2013 a je doprovázena jednou z největších změn historie iOS co se vzhledu a bezpečnosti týče. Mezi nejvýznamnější změny patří přidání menu Control Center poskytující rychlý přístup k některým funkcím telefonu, jako je např. režim letadlo, zapnutí sítě Wifi nebo umožňující rychlé spuštění aplikací jako je fotoaparát nebo kalkulačka. Z hlediska bezpečnosti je ve verzi zpřístupněna možnost povolit nebo zakázat jednotlivá oprávnění přístupu k citlivým informacím individuálně pro každou aplikaci. Dalším vylepšením verze je podpora služby Airdrop pro rychlé sdílení fotografií, videí a jiných souborů s lidmi v okolí.

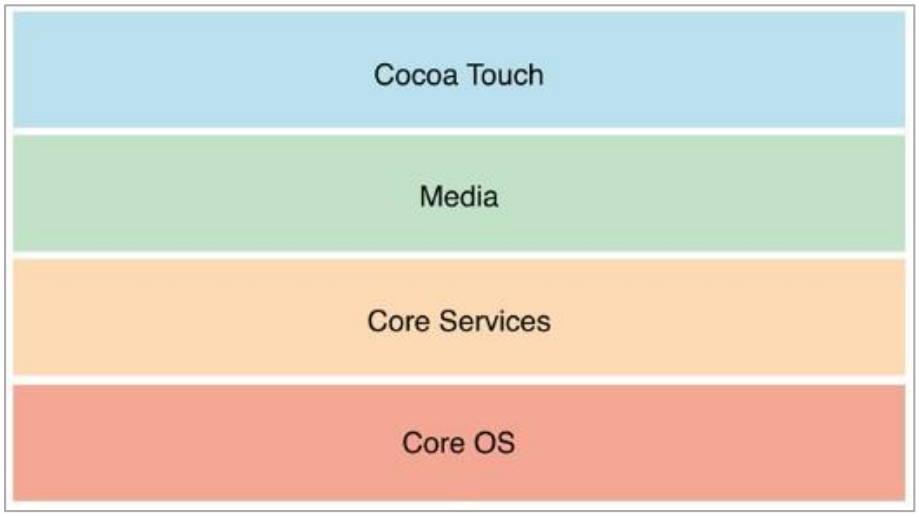
3.2 Základní architektura OS iOS

Platforma iOS je upravenou verzí operačního systému Mac OS X[32], který společnost Apple používá ve svých osobních počítačích. Skládá se ze čtyř základních vrstev, které jsou přizpůsobeny tak, aby poskytovaly mnohem větší uzavřenost a bezpečnost systému, než je tomu u systému Android. Vrstvy komunikují pomocí rozhraní prostřednictvím tzv. frameworks neboli rámců. Tyto rámce jsou u každé vrstvy dostupné pomocí specifických

[31] iCloud je cloudová služba poskytována společností Apple Inc. pro ukládání hudby, iOS aplikací nebo dat na vzdálené počítačové servery. Služba slouží také jako synchronizační centrum pro synchronizaci fotografií, kontaktů, e-mailů, kalendářů, poznámek nebo záložek jednotlivých zařízení.

[32] Mac OS X je původní označení operačního systému pro počítače Macintosh vyvíjeným společností Apple Inc. V současné době je nahrazen svou aktuální verzí pod označením OS X.

balíčků pro každou vrstvu. Na následujícím obrázku je zobrazena základní architektura složená z jednotlivých vrstev.



Obrázek 6 Model vrstev OS iOS [14]

3.2.1 Core OS (Jádro systému)

Vrstva jádra je nejnižší vrstvou architektury, pomocí které jsou zpřístupněny nízkoúrovňové funkce, prostřednictvím kterých jsou poskytovány jednotlivé technologie. Tato vrstva komunikuje přímo s hardwarem zařízení a obstarává základní služby, jako je správa paměti, souborový systém, šifrování a dešifrování nebo poskytování přístupu k externímu příslušenství. Mezi významné rámce z hlediska bezpečnosti patří tzv. Security Framework, což je rámec zodpovědný za bezpečné nakládání s citlivými daty [15].

3.2.2 Core Services (Základní služby)

Core Services je vrstva poskytující základní systémové služby, které využívají všechny ostatní aplikace. Vrstva obsahuje základní rozhraní pro sdílení a přístup k souborům prostřednictvím softwaru iTunes[33] nebo v rámci služby iCloud. Z hlediska bezpečnosti je

[33] iTunes je software nabízený zdarma společností Apple Inc., který slouží nejen pro správu mobilních zařízení, ale i jako prohlížeč multimediálního obsahu. Prostřednictvím tohoto programu se lze také připojit k internetovému obchodu iTunes Store.

tato vrstva zodpovědná za služby, které se využívají k realizaci ukládání dat a k realizaci kryptografických funkcí využívaných v rámci keychain[34] databáze [15].

3.2.3 Media Layer (Vrstva médií)

Vrstva médií obsahuje podporu pro grafické, audio a video technologie, které jsou využívány pro vytváření složitějších multimediálních aplikací. Vrstva je dále schopna přehrávat a nahrávat vysoce kvalitní zvuk a umožňuje využívání vibračních funkcí, pokud jsou zařízením podporovány.

3.2.4 Cocoa Touch (Dotyková vrstva)

Poslední vrstvou architektury je dotyková vrstva, která obsahuje jedny z nejdůležitějších frameworků používající se při tvorbě aplikací. Prostřednictvím této vrstvy je umožněna vzájemná interakce zařízení s uživatelem pomocí dotykového ovládání nebo pomocí rozpoznání předem nadefinovaných gest. Vrstva dále zprostředkovává podporu multitaskingu a push notifications.

3.3 Bezpečnostní prvky systému

Zařízení s operačním systémem iOS jsou postaveny tak, aby udržovaly vysoký stupeň bezpečnosti, a to i v případě minimálních uživatelských zkušeností. Z tohoto důvodu jsou mnohé bezpečnostní funkce povoleny již ve výchozím nastavení, čímž pro uživatele odpadá nutnost provádět složitá konfigurační nastavení [16].

Podobně jako u systému Android lze bezpečnostní model systému iOS rozdělit do těchto základních kategorií:

- System Security (Bezpečnost systému)
- Encryption and Data Protection (Šifrování a ochrana dat)
- Apps Security (Bezpečnost aplikací)
- Update (Aktualizace)

[34] Keychain neboli klíčenka je souborový systém implementovaný do systému v rámci SQL (Structured Query Language) databáze. Souborový systém slouží jako bezpečný prostor pro ukládání klíčů nebo přihlašovacích údajů.

3.3.1 System Security (Bezpečnost systému)

Mezi zásadní bezpečnostní prvky systému patří:

- Password (Heslo)
- User Profiles (Uživatelské profily)
- Memory Management Security (Bezpečnost správy paměti)

Password (Heslo)

Ochrana přístupu k zařízení pomocí uživatelského hesla je i v tomto případě velmi důležitým bezpečnostním prvkem, který může ovlivnit každý uživatel sám. Heslo se vkládá na uzamčené obrazovce tzv. Lockscreen a v případě, že není vloženo správně, systém odmítne uživateli do zařízení přístup. Po každém neúspěšném zadání hesla se možnost jeho opětovného vložení časově prodlužuje, což rapidně snižuje úspěšnost pomocí útoků hrubou silou. Uživatel si navíc může nastavit, aby se po deseti neúspěšných pokusech o zadání hesla ze zařízení smazala veškerá data.

User Profiles (Uživatelské profily)

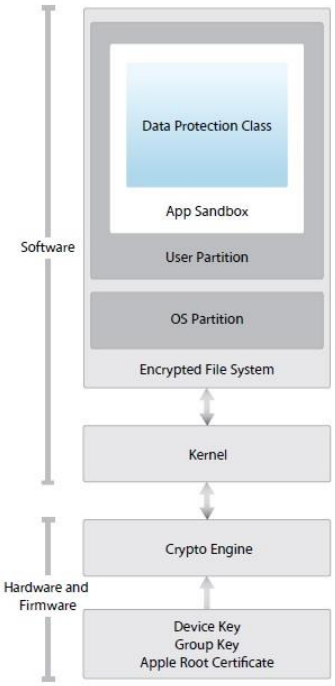
Další neocenitelným prvkem kontroly nad zařízením je možnost vytváření uživatelských profilů. Tuto možnost ocení především majitelé firem v rámci své bezpečnostní politiky. Princip spočívá v tom, že prostřednictvím programu iPCU (iPhone Configuration Utility) nainstalovaném v systému Windows, může správce IT vytvářet a upravovat důležitá nastavení vybraných skupin zařízení. Tyto předvolby se nastavují pomocí uživatelských profilů, které se ukládají v souboru XML (Extensible Markup Language) a lze jej zálohovat pro pozdější použití. Profily obsahují informace např. o konfiguraci sítě VPN, nastavení sítě Wifi, nastavení poštovních účtů nebo certifikátů. Vhodně nastavené zařízení může uživateli odepřít celou řadu povolení, jako např. povolení instalovat aplikace nebo měnit nastavení síťového provozu [16]. Správa zařízení prostřednictvím programu iPCU je podrobněji vysvětlena v poslední kapitole praktické části práce.

Memory Management Security (Bezpečnost správy paměti)

U systému Android byla popsána technologie správy paměti ASLR a její následné vylepšení technologií PIE, které jsou i v případě systému iOS velmi důležitým prvkem pro ochranu systému před škodlivými kódy jako jsou exploity. U systému iOS byly tyto technologie poprvé zpřístupněny ve verzi iOS 4.3.

3.3.2 Encryption and Data Protection (Šifrování a ochrana dat)

Obdobně jako systém Android využívá i systém iOS k ochraně dat různé kryptografické funkce. Architektura systému je z tohoto hlediska rozdělena na dvě vrstvy využívající hardwarové a softwarové technologie (viz. Obrázek 7).



Obrázek 7 Bezpečnostní model systému iOS [16]

Hardware Encryption (Hardwarové šifrování)

Od uvedení zařízení iPhone 3GS je v iOS implementován tzv. AES 256 bitový šifrovací stroj (Crypto Engine) založený na principu DMA[35] (Direct Memory Access), který je zabudovaný mezi hlavní systémovou pamětí a pamětí flash. Tento stroj pracuje spolu s SHA-1 kryptografickou funkcí na hardwarové úrovni. Pomocí jedinečného identifikátoru zařízení UID (Unique Identifier) a identifikátoru skupiny zařízení GID (Group Identifier) vytváří stroj AES 256 bitový klíč, který je vytaven do křemíku procesoru již při jeho výrobě, což zaručuje, že s ním nemůže být nijak manipulováno a nemůže k němu být přistupováno jinak nežli prostřednictvím šifrovacího stroje. Data přitom nelze přečíst ani

[35] Přímý přístup do paměti DMA umožňuje přenos dat mezi operační pamětí a výstupními zařízeními bez asistence centrální procesorové jednotky CPU (Central Processing Unit).

za pomocí speciálního softwaru nebo firmwaru. V tomto případě lze číst pouze výsledky šifrovacích a dešifrovacích operací. Při vytváření šifrovacích klíčů je i zde využito generátoru náhodných čísel RNG (Random Number Generation). Výsledkem celé této sekvence šifrování je, že v zařízení jsou šifrována všechna data bez výjimky a tato data jsou kryptograficky vázána ke konkrétnímu přístroji a nemůžou být dekódovány v žádném jiném zařízení [16] [27].

Hardwarové šifrování má význam v případě, pokud by se útočník ze zařízení pokusil odstranit paměť flash a tato data následně přečíst. V takovém případě by jeho pokus byl neúspěšný, jelikož se data vážou pouze ke konkrétnímu zařízení. Dalším důvodem je možnost použití funkce Remote Wipe, která umožňuje ze zařízení vymazat na dálku veškerá data během několika sekund. Této rychlosti lze dosáhnout rychlým odstraněním šifrovacího klíče a nikoliv zdlouhavým mazáním veškerého obsahu.

Princip hardwarového šifrování je přitom velmi efektivním způsobem s minimálním dopadem na výkon procesoru. To je v případě mobilních zařízení velmi důležité z hlediska výdrže baterie.

Data Protection (Ochrana osobních údajů)

Ochrana osobních dat je dodatečnou ochranou a je k dispozici pouze pro zařízení, které umožňují hardwarové šifrování. Aktivuje se automaticky při nastavení vstupního hesla. Prostřednictvím této ochrany jsou chráněny osobní údaje v souborech vybraných aplikací z důvodu, aby zařízení mohlo v uzamčeném režimu reagovat na některé události jako je např. příchozí hovor nebo zpráva a aniž by musela být dešifrována veškerá data. Funkce je přístupná pouze v aplikacích, které byly vývojáři speciálně navrženy prostřednictvím programovacího rozhraní API [16].

Další důležitou ochranou je ochrana síťového provozu. To znamená, že systém zabezpečuje veškerou komunikaci směřující ven i dovnitř zařízení. K tomu používá řadu standardních síťových protokolů využívaných pro autentizaci, autorizaci a šifrovanou komunikaci. Aplikace přeinstalované v systému jako je iMessage, FaceTime nebo Push Notifications jsou zabezpečeny právě tímto způsobem [16]. K vytvoření takové komunikace systém využívá protokoly SSL a TSL (Transport Layer Security). Tyto protokoly jsou vloženy mezi transportní a aplikační vrstvu a poskytují možnost vytvoření bezpečné komunikace pro internetové služby, jako je např. internetové bankovnictví nebo elektronická pošta. Systém dále umožňuje zabezpečení síťové komunikace Wi-Fi nebo

Bluetooth za pomocí standardních protokolů a vytvoření privátní sítě VPN. V případě sdílení dat přes rozhraní bluetooth je tato možnost omezena pouze na zařízení společnosti Apple.

3.3.3 Apps Security (Bezpečnost aplikací)

Pro ochranu na aplikační úrovni využívá systém iOS tato bezpečnostní opatření:

- Sandboxing (Izolovaný prostor)
- Application Signing and Verification (Podpis aplikací a ověření)

Sandboxing (Izolovaný prostor)

Všechny aplikace třetích stran běží i v systému iOS v izolovaném prostoru zvaném Sandbox. To znamená, že nejsou oprávněny sdílet informace s jinými aplikacemi, přistupovat k jádru systému nebo získat práva administrátora. Pracují tedy v prostředí, kde je zaručena dokonalá separace mezi aplikacemi navzájem i mezi systémem. Aplikaci není ani dovoleno, aby zjistila přítomnost aplikace jiné. Pokud však aplikace pro svoji činnost některou z těchto informací vyžaduje, musí jí vývojář předem nastavit patřičná oprávnění prostřednictvím programovacího rozhraní API [27].

Application Signing and Verification (Podpis aplikací a ověření)

Aby bylo zajištěno, že aplikace pochází ze známého zdroje a neobsahuje žádný škodlivý kód, vyžaduje systém před jejím načtením, aby byla digitálně podepsána obdobně jako v případě systému Android. Taková aplikace je potom podepsána certifikátem vydaným společností Apple. Rozdíl je však ten, že aby mohli vývojáři vyvíjet a poskytovat své aplikace pro systém iOS, musí být nejprve registrování v rámci tzv. iOS Developer Program[36]. V takovém případě musí každý vývojář zaplatit registrační poplatek 99 $ za rok, načež bude ověřena jeho totožnost a bude mu vydán certifikát společnosti Apple. Tento certifikát mu umožní podepisovat své aplikace, které může následně distribuovat v obchodě App Store. Tento princip je bezpečný z hlediska toho, že v případě, kdy se objeví škodlivá aplikace, je ihned dohledán vývojář, který tuto aplikaci poskytnul. Každá

[36] iOS Developer Program slouží pro registraci vývojářů u společnosti Apple Inc., kteří po splnění stanovených podmínek mohou vyvíjet a distribuovat své produkty.

aplikace je navíc ještě před svým uvedením na trh důkladně otestována pracovníky Apple a až potom je umožněna její distribuce [16].

Další možností, jak vytvářet a poskytovat své aplikace, je členství a registrace v programu IDEP (iOS Developer Enterprise Program). Jedná se o členství, které se vztahuje na firmy a organizace, které mohou v rámci své organizace vytvářet a distribuovat aplikace do zařízení svých zaměstnanců. V tomto případě musí organizace zaplatit registrační poplatek 299 $ za rok [16].

Na rozdíl od systému Android, systém iOS neumožňuje uživatelům instalovat nepodepsané aplikace z neznámých zdrojů.

3.3.4 Update (Aktualizace)

Apple vydává aktualizace systému pravidelně a pro všechna podporovaná zařízení současně. Řeší v nich jak drobné úpravy samotného uživatelského prostředí, tak především otázky týkající se bezpečnosti. Uživatel je na ně upozorněn přímo na svém zařízení nebo prostřednictvím programu iTunes. V případě vydání nové aktualizace jí je možné instalovat prostřednictvím tohoto programu nebo vzduchem s využitím technologie OTA. V obou případech je před instalací aktualizačního balíčku vyžádán dotaz pro ověření na server Apple. Tím je zajištěna autentizace a integrita celého instalačního procesu. V případě, že server instalaci povolí, je proces instalace spuštěn.

Mezi prvky které doprovází proces aktualizace a spouštění systému iOS patří:

- Secure Boot Chain (Bezpečný spouštěcí proces)
- System Software Personalization (Systémová softwarová personalizace)

Secure Boot Chain (Bezpečný spouštěcí proces)

Každý krok ve spouštěcím procesu obsahuje komponenty, které jsou kryptograficky podepsané, a pokračuje pouze po jejich ověření prostřednictvím serveru Apple. Tímto způsobem se od každého zapnutí přístroje ověřují jednotlivé kroky spouštěcího procesu až do doby, než je systém spuštěn. Toto bezpečnostní opatření zajišťuje, že nejnižší vrstvy systému nejsou porušeny a zároveň umožňuje spustit systém pouze na ověřených zařízeních Apple. V případě, že některý krok tohoto procesu nelze ověřit, je spouštěcí

proces zastaven a na zařízení se zobrazí obrazovka vyžadující připojení k iTunes (tzv. Recovery Mode) nebo je zařízení uvedeno do DFU[37] (Device Firmware Update) módu. V obou případech je nutné zařízení připojit k programu iTunes přes kabel USB a provést obnovení do továrního nastavení [16] [27]. Procesem bezpečného spouštění systému se zabývá kapitola 5.4 obsažena v praktické části práce.

System Software Personalization (softwarová Systémová personalizace)

V předchozím případě bylo řečeno, že na zařízení může být instalován a spouštěn pouze systém, který je ověřený prostřednictvím serveru Apple. Existuje však varianta, že by se útočník mohl pokusit instalovat starší verzi operačního systému. Tímto způsobem by byl schopen vyřadit z provozu některé bezpečnostní prvky, které obsahují pouze verze novější. Takovému způsobu, kdy je na zařízení instalována starší verze operačního systému, se říká downgrade a má mu zabránit proces zvaný System Software Personalization. V tomto případě si iTunes vyžádá z vašeho zařízení jedinečné číslo ECID[38] (Electronic Chip ID) a toto číslo zašle na podpisový server Apple. Ten ověří, zda může být proveden restore na danou verzi systému nebo nikoli. Záleží na tom, zda Apple daný firmware ještě podporuje [16].

[37] DFU mód uvádí zařízení do stavu, kdy je možné v něm zcela přehrát stávající firmware na rozdíl od Recovery módu, kdy se zařízení pouze aktualizuje. Při Recovery módu je ochrana prostřednictvím *System Software Personalization* aktivní. Při DFU módu je tato ochrana vypnutá.

[38] Jedinečné a unikátní číslo ECID mají v sobě zařízení řady iPhone 3gs a vyšší. Číslo je obsaženo v chipu zařízení a prostřednictvím programu iTunes se ověřuje, jestli lze provést restore na danou verzi firmwaru či nikoli.

4 OPERAČNÍ SYSTÉM WINDOWS PHONE 8

Předchůdcem řady operačních systémů Windows Phone byl operační systém Windows Mobile, který byl založen na architektuře jádra Windows CE[39] (Windows Embedded CE). Systém byl vyvíjen společností Microsoft a byl určen pro mobilní zařízení, jako jsou smartphony nebo PDA. Vývoj systému byl postupně ukončen s příchodem nové generace operačního systému pod názvem Windows Phone 7, který byl také vyvíjen společností Microsoft. Tento systém, přestože je postaven na stejném jádře Windows CE, nebyl s předchozí generací zpětně kompatibilní. V současné době je k dispozici nejnovější řada tohoto systému pod názvem Windows Phone 8, která je postavena na zcela nové architektuře.

4.1 Historie

Od roku 2010 byly do současnosti představeny celkem dvě základní verze systému Windows Phone. Jedná se o verzi Windows Phone 7 a verzi Windows Phone 8.

Windows Phone 7

První verze systému byla zpřístupněna v říjnu roku 2010. Verze přinášela zcela odlišné uživatelské rozhraní, kdy se ze systému vytratily klasické ikony, které nahradila tzv. centra uspořádaná do dlaždic. Toto uživatelské rozhraní bylo implementováno také v operačním systému Windows 8 a v informatice se označuje jako rozhraní Metro. V roce 2011 vydal Microsoft první aktualizovanou verzi pro tento systém s názvem Windows Phone NoDo, která přinesla řadu vylepšení. Mezi tato vylepšení patří především integrovaná funkce kopírování a vkládání textů, rychlejší proces spouštění aplikací s celkovou optimalizací výkonu a integrace sociální služby Facebook. Další aktualizovaná verze byla vydána v září roku 2011 pod názvem Windows Phone 7 Mango. V této verzi byl prvně implementován multitasking pro aplikace třetích stran a nová verze prohlížeče Internet Explorer 9. Dalším významným vylepšením byla integrace služeb jako je Twitter, Windows Live Messenger

[39] Windows CE je modulární operační systém sloužící jako základ pro více druhů zařízení. Systém byl vyvinut společností Microsoft a má odlišnou architekturu jádra oproti klasickému operačnímu systému Microsoft Windows. Hlavní využití tohoto systému je u zařízení, která mají málo místa pro uložení operačního systému. V takovém případě stačí několik jednotek megabajtů.

nebo Linkedln. Poslední aktualizovanou verzí byla verze Windows Phone 7.8 Tango, která byla vydána v červnu roku 2012. Jedná se o verzi, která byla předchůdcem verze Windows Phone 8. Přestože s touto verzí není zpětně kompatibilní, přináší některé funkce, které jsou prezentovány právě v této verzi. Jedná se především o vylepšení uživatelského rozhraní úvodní obrazovky, na které jsou podporována centra v podobě aktivních oken. U těchto oken lze měnit jejich velikost a lze v nich zobrazovat aktuální informace v reálném čase [23].

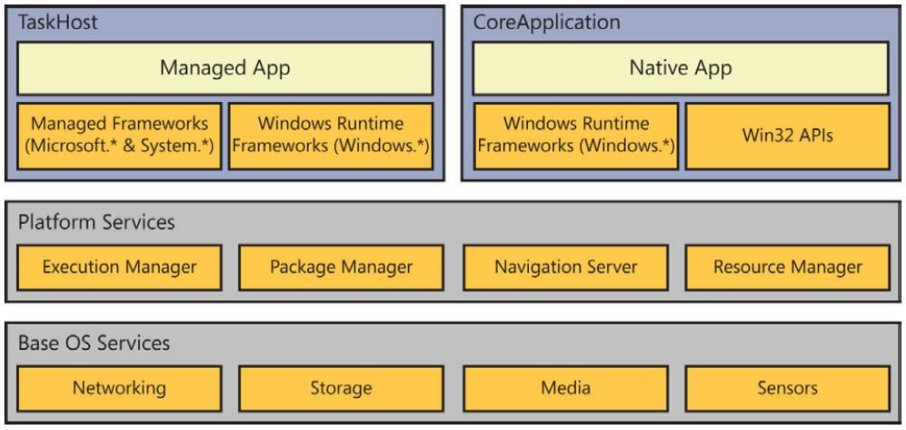
Windows Phone 8

První verze druhé generace operačního systému Windows Phone byla vydána v říjnu roku 2012 pod označením Windows Phone 8 Apollo. Zásadní změnou této generace je přechod na upravené jádro Windows NT, které je využíváno také systémem Windows 8. Z tohoto důvodu nemohou zařízení vyvinutá pro Windows Phone 7 provést upgrade na tuto verzi ani používat její aplikace. Hlavní výhodou je však možnost využití vícejádrových procesorů, použití multitaskingu na pozadí, vzájemná kompatibilita s aplikacemi vytvořenými pro systém Windows 8, podpora HD rozlišení 1280 x 720 a podpora pro paměťové karty MicroSD [24].

V současné době je pro tento systém k dispozici nejnovější aktualizace 3, která přináší mimo jiná vylepšení také podporu Full HD rozlišení 1980 x 1050, novou správu datového úložiště, nová vylepšení pro uživatele se zrakovým postižením a sdílení internetu přes rozhraní bluetooth nebo Wi-Fi.

4.2 Základní architektura OS Windows Phone 8

Základní architektura systému se člení do čtyř základních vrstev, kdy každá z vrstev poskytuje určité služby systému. Tyto služby jsou zobrazeny na obrázku 8.



Obrázek 8 Model vrstev OS Windows Phone 8 [17]

4.2.1 Base OS Services (Základní služby OS)

Přístup k základním technologiím systému poskytují služby na úrovni jádra. Základ jádra je přitom shodný s architekturou systému Windows 8 a dělí se na dvě základní složky:

- Windows Core System (Jádro systému Windows)
- Mobile Core (Mobilní jádro)

Windows Core System (Jádro systému Windows)

Jádro systému Windows poskytuje základní služby jako je souborový systém NTFS[40] (New Technology File System), služby pro jádro NT (New Technology) a bezpečnostní a síťové služby.

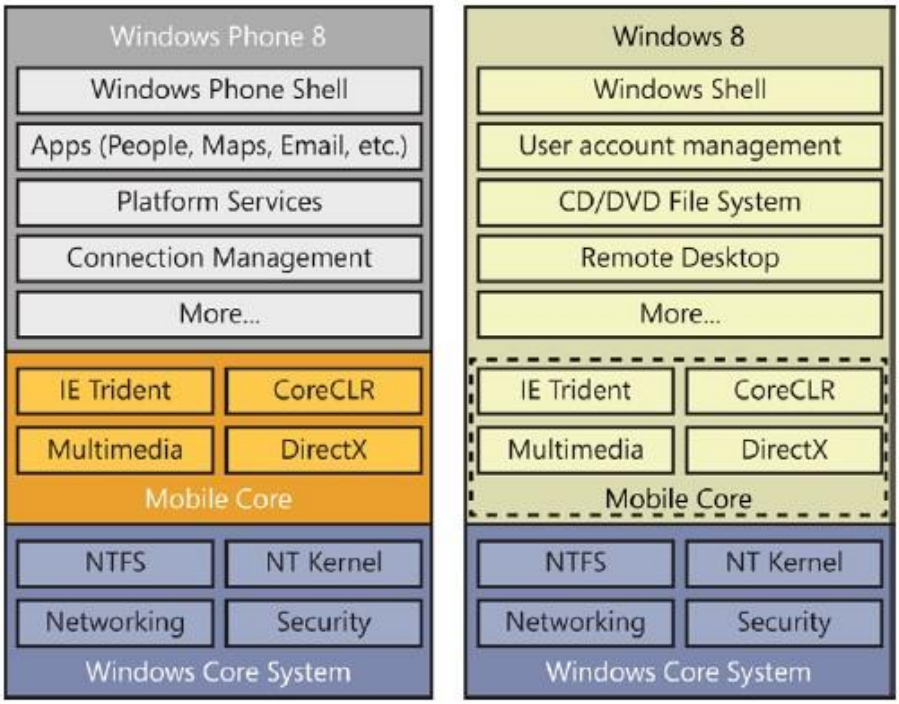
Mobile Core (Mobilní jádro)

Složka mobilního jádra není součástí základní složky jádra, jako je tomu v případě architektury Windows 8, ale je oddělená a pro mobilní zařízení velmi důležitá. Zahrnuje v sobě komponenty, jako jsou multimédia, IE Trident[41], DirectX nebo CoreCLR[42]. Celá

[40] NTFS souborový systém byl vyvinut společností Microsoft jako nativní souborový systém pro platformu Windows NT. Oproti zastaralému souborovému systému FAT (File Allocation Table) umožňuje spoustu nových funkcí, jako je např. šifrování nebo komprese dat souborového systému nebo možnost přidělování práv k souborům.

[41] IE Trident je renderovací jádro používané v prohlížeči Internet Explorer, prostřednictvím kterého mohou vývojáři snadno přidat do svých aplikací některé funkce pro prohlížení webových stránek.

architektura jádra je zobrazena na obrázku 9 spolu s platformou operačního sytému Windows 8. Zde jsou vidět také některé složky sdílené oběma systémy. Všimněme si, že v případě systému Windows 8 obsahuje mobilní jádro stejné komponenty, které však nejsou oddělené, ale jsou součástí většího celku funkcí [17].



Obrázek 9 Jádro systému Windows Phone 8 a Windows 8 [17]

4.2.2 Platform Services (Platforma služeb)

Další vrstvou architektury je vrstva služeb platformy. Tato vrstva zahrnuje tyto čtyři služby[17]:

- Package Manager (Správce balíčků)
- Execution Manager (Výkonný manažer)
- Navigation Server (Navigační server)
- Resource Manager (Správce zdrojů)

Package Manager (Správce balíčků)

---

[42] CoreCLR je omezená verze plnohodnotného základního běhového prostředí pro spouštění řízeného kódu CLR (Common Language Runtime), jenž se stará o běh programu v systémech Windows.

Správce instalačních balíčků je zodpovědný za instalaci/odinstalaci aplikací a zachování veškerých metadat, které si aplikace vytvoří během své životnosti a to včetně licencí. Další jeho významnou úlohou je udržení informací týkajících se aplikačních oken, která může mít uživatel nastavené na úvodní obrazovce [17].

Execution Manager (Výkonný manažer)

Výkonný manažer se stará o veškerou logiku spojenou s realizací života aplikace. Jinými slovy vytváří proces, ve kterém aplikace běží a vyvolává události spojené s jejím startem, vypnutím nebo deaktivací. Podobné úlohy vytváří i pro aplikace běžící na pozadí [17].

Navigation Server (Navigační server)

Navigační server je zodpovědný za veškerý pohyb mezi aplikacemi pracujícími na popředí. V případě, že aktivujeme aplikační okno na úvodní obrazovce, navigační server předá tuto informaci výkonnému manažeru, a ten provede požadovanou instrukci [17].

Resource Manager (Správce zdrojů)

Správce zdrojů sleduje využívání systémových zdrojů a to zejména paměti nebo procesoru. Tím je zaručeno, že práce se zařízením je vždy rychlá. Pokud některá z aplikací nebo některý z procesů běžících na pozadí překročí své přidělené zdroje, bude jeho činnost ukončena [17].

4.2.3 TaskHost a CoreApplication

Nejvyšší vrstva architektury zahrnuje dvě skupiny aplikačních modelů. Prvním je TaskHost představující aplikační model XAML (Extensible Application Markup Language) jenž byl hlavním modelem od uvedení systému Windows Phone 7 a v tomto systému je zahrnut z důvodu kompatibility. Druhým je CoreApplication, který je novým aplikačním modelem pro verzi Windows Phone 8. Oba tyto modely jsou vývojáři využívány při tvorbě aplikací [17].

4.3 Bezpečnostní prvky systému

Operační systém Windows Phone 8 je svou architekturou od předešlých dvou systémů do jisté míry odlišný. To se týká i jeho bezpečnosti, která vychází z architektury jádra systému Windows NT a ze softwarové stránky novějších systémů Windows.

Mezi základní skupiny z hlediska bezpečnosti patří:

- System Security (Bezpečnost systému)
- Encryption a Data Protection (Šifrování a ochrana dat)
- Apps Security (Bezpečnost aplikací)
- Update (Aktualizace)

4.3.1 System Security (Bezpečnost systému)

Společnost Microsoft využívá u systému Windows Phone 8 podobné bezpečnostní prvky jako Apple u iOS. Mezi tyto prvky patří zejména:

- Password (Heslo)
- Secure Boot Chain (Bezpečný spouštěcí proces)
- Mobile Device Management (Správa mobilních zařízení)

Password (Heslo)

I v tomto případě je možné využít aktivaci přístupového hesla zařízení a uživatelem nastavit přihlašovací heslo. V případě, že si uživatel nenastaví heslo sám, mohou správci IT v rámci bezpečnostní politiky uživatelům zadání hesla nařídit. To lze učinit prostřednictvím vzdáleného přístupu EAS[43] (Exchange Active Sync). V případě, že by bylo zařízení ztraceno nebo odcizeno, lze prostřednictvím vzdáleného serveru a aplikace *Outlook Web App* ze zařízení smazat veškerá data případně zařízení lokalizovat. V tomto případě je však vyžadována registrace na webových stránkách (www.windowsphone.com) [18].

Secure Boot Chain (Bezpečný spouštěcí proces)

Při zavádění systému používá Microsoft obdobný spouštěcí proces, který se používá u systému iOS. Tento proces s podepisováním kódu pomáhá zajistit integritu celého systému včetně aplikací, což jej chrání před škodlivým malwarem, a to zejména před rootkity. Ochrana zavádění firmwaru je zavedena do zařízení již při jeho výrobě a znamená, že všechny binární soubory při načtení musí být podepsány důvěryhodnou autoritou.

[43] EAS je zkratkou pro aplikační protokol založený na formátu (XML), který je vyvinutý společností Microsoft. Protokol je určen pro bezdrátovou synchronizaci dat jako e-maily, poznámky, kalendáře, úkoly, kontakty nebo jiné zprávy zasílané ze serveru na mobilní zařízení.

K tomuto účelu je využíván nový standard rozhraní UEFI[44] (Unified Extensible Firmware Interface). Výsledkem tohoto procesu je, že veškerý kód v operačním systému je podepsaný certifikátem včetně ovladačů i aplikací [18].

Mobile Device Management (Správa mobilních zařízení)

MDM (Mobile Device Management) je zkratkou pro správu mobilních zařízení. To umožňuje správcům IT převzít kontrolu nad vybranými zařízeními a nastavit pravidla při jejich používání. V takovém případě lze dálkově provádět konfigurační nastavení zařízení, provádět distribuci aplikací nebo v případě ztráty provést uzamknutí nebo vymazání zařízení na dálku. Organizace pro tuto správu mohou používat služby, jako je např. Windows Intune, kterou provozuje společnost Microsoft nebo služby třetích stran, jako je AirWatch nebo Zenprise [18].

4.3.2 Encryption a Data Protection (Šifrování a ochrana dat)

Pro šifrování obsahu uloženého v interní paměti využívá systém nástroj BitLocker[45] s šifrovacím algoritmem AES 128bit. Toto šifrování lze povolit buďto pomocí vzdáleného přístupu EAS, nebo přímo v zařízení. V případě, že je povoleno, bude systém šifrovat veškerý obsah uložený na interním úložišti. Šifrovací klíč je přitom chráněn pomocí TPM[46] (Trusted Platform Module), který je vázán na UEFI Secure Boot Proces, což zajistí jeho ochranu. Tento princip je obdobný jako v případě ochrany spouštěcího procesu systému iOS [18].

Pro další rozšíření ukládacího prostoru systém umožňuje využít externích karet MicroSD. V tomto případě však umožňuje uživatelům ukládat pouze mediální soubory, jako jsou

---

[44] UEFI je nový standard sloužící jako vylepšená náhrada za zastaralé firmwarové rozhraní BIOS (Basic Input Output System). V současné době je také dostupný u základních desek novějších počítačových systémů.

[45] BitLocker je nástroj přístupný od verze systému Windows 8, který umožňuje šifrování systémového disku, popřípadě šifrování jiných úložišť, jako jsou např. flash disky. Účelem tohoto šifrování je poskytnout větší ochranu dat v případě ztráty nebo odcizení zařízení obdobně, jako je tomu v případě systému iOS.

[46] TPM je zkratka pro modul mikroprocesoru určeného pro ukládání artefaktů obsahující hesla, certifikáty, šifrovací klíče, které jsou využívány pro ověření důvěryhodnosti platformy.

obrázky, videa nebo hudba. Tato data však zatím není v této verzi systému umožněno šifrovat.

Další výhodou systému je, že jako jediný systém pro mobilní zařízení nabízí podporu technologie IRM (Information Rights Management), která chrání informace před neoprávněným přístupem. To znamená, že pokud je tato technologie přístupná, jsou data v dokumentech nebo e-mailech šifrována a mohou je tak vidět pouze autorizovaní uživatelé. IRM se může využít také k omezení některých práv dokumentů, jako je např. omezení přístupu pouze pro čtení, zákaz kopírování nebo zákaz tisku. Tyto funkce včetně zákazu distribuce zabraňují možnému úniku informací k neoprávněným osobám [18].

Pro zabezpečení síťové komunikace mezi zařízením a webovým serverem systém používá připojení SSL, které je šifrováno pomocí 128 bit nebo 256 bit AES protokolu. S příchodem nové verze systému Windows Phone 8.1 bude možné připojit zařízení do VPN sítě nebo využívat platebních transakcí pomocí NFC technologie. Dále bude možné využívat mimo základní bezpečnostní protokoly Wifi také podporu Wifi Direct, která slouží pro přímé propojení dvou zařízení bez nutnosti použití routeru. Ve všech těchto případech bude probíhající komunikace šifrována.

4.3.3 Apps Security (Bezpečnost aplikací)

Podobně jako u předchozích dvou systémů využívá i systém Windows Phone 8 pro ochranu aplikací tyto bezpečnostní prvky:

- Sandboxing (Izolovaný prostor)
- Application Signing a Verification (Podpis aplikací a ověření)

Sandboxing (Izolovaný prostor)

Také v tomto případě musí mít každá aplikace přidělena zvláštní oprávnění, aby mohla přistupovat k datům, která jsou vyhrazená pro aplikaci jinou. Tato oprávnění jsou deklarovaná v souboru aplikačního manifestu. Všichni vývojáři mají tak povinnost upozornit uživatele při instalaci každé aplikace na tato oprávnění.

Běh v izolovaném prostředí se využívá i pro webový prohlížeč Internet Explorer 10, který navíc nepodporuje doplňkové moduly (pluginy), skriptovací jazyk JavaScript nebo aktivní prvky ActiveX. Toto omezení rapidně snižuje možné útoky. V systému je dále implementována technologie SmartScreen, která varuje uživatele před webovými stránkami, jenž jsou označeny jako škodlivé [18].

Application Signing a Verification (Podpis aplikací a ověření)

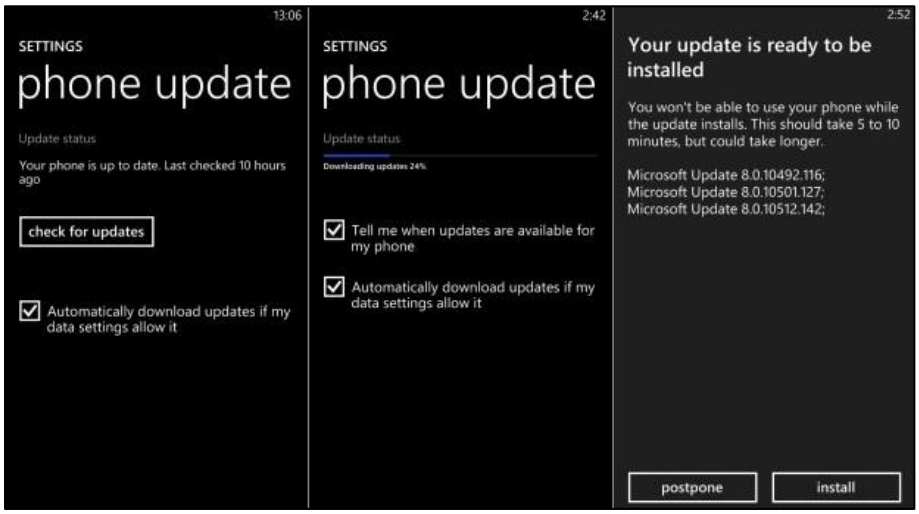
V systému musí být všechny aplikace podepsané certifikáty. Pro získání aplikací slouží dva možné zdroje. Prvním zdrojem je internetový obchod Windows Phone Store (www.windowsphone.com/store) provozován společností Microsoft. Zde jsou všechny aplikace předem kontrolovány a testovány na škodlivý software a po úspěšné kontrole jsou opatřeny certifikátem. Druhým způsobem je možnost opatření si aplikace z firemních stránek důvěryhodné společnosti. Obdobně jako v případě systému iOS musí společnost, která chce distribuovat své vlastní aplikace, projít registračním procesem a vytvořit si tak firemní účet na stránkách Windows Phone Dev Center (http://dev.windowsphone.com). Jakmile splní všechny registrační požadavky, je jí vydán firemní certifikát [18].

Certifikáty ve Windows Phone 8 jsou používány především k [18]:

- Vytvoření bezpečného kanálu mezi zařízením a webovým serverem pomocí protokolu SSL.
- Ověření uživatele v rámci protokolu EAS.
- Ověření licencí pro instalaci aplikací v rámci obchodu Windows Phone Store nebo distribučního místa vlastní společnosti.

4.3.4 Update (Aktualizace)

Aktualizace jsou zákazníkům dodávány zdarma prostřednictvím instalačního balíčku služby OTA. V takovém případě však musí mít uživatel v zařízení tuto možnost povolenou. V případě, že ji povolenu má, bude zařízením upozorněn, že je k dispozici nová aktualizace. Poté stačí jen potvrdit instalaci a aktualizace se po ověření do zařízení nainstaluje. Pokud si chce uživatel zkontrolovat dostupnost nejnovější aktualizace sám, lze tak učinit přímo v zařízení v záložce nastavení (viz. Obrázek 10).



Obrázek 10 Aktualizace systému Windows Phone 8 [25]

Z důvodu, aby byl systém co možná nejlépe vyladěn a aby bylo odstraněno co nejvíce bezpečnostních hrozeb, využívá společnost Microsoft vývojový proces zvaný SDL (Security Development Lifecycle). V tomto případě je za pomocí inženýrských týmů společnosti prováděno rozsáhlé modelování hrozeb, penetrační testování a další procesy, které pomáhají zabránit k neoprávněnému přístupu ke zdrojům zařízení a na jejich základě jsou vydávány aktualizační balíčky [18].

II. PRAKTICKÁ ČÁST

5 TEST DETEKCE MALWARU

V praktické části se zaměříme především na operační systém Android, který je ze všech tří systémů nejvíce zranitelný. V první části kapitoly si vyzkoušíme, zda je systém schopen detekovat škodlivou aplikaci jak při pokusu o její instalaci, tak v případě, kdy je aplikace pouze nahrána v úložišti zařízení. V další části se zaměříme na možné způsoby, jak se přesvědčit, zda je aplikace opravdu škodlivá či nikoli. K tomuto účelu využijeme prostředí některých vybraných webových služeb. V poslední části kapitoly si podrobněji popíšeme bezpečný spouštěcí proces systému iOS.

5.1 Výběr hardwaru a softwaru pro testování

Důležitým krokem praktické části je volba vhodného virtualizačního softwaru, který bude nainstalován a spuštěn na dostatečně výkonném hardwaru. V současné době je na trhu celá řada volně dostupných programů, které jsou pro nekomerční a testovací účely poskytovány zpravidla bezplatně. Pro tuto práci jsem použil notebook s označením ProBook 4720s od výrobce Hewlett Packard. Toto zařízení disponuje dvoujádrovým procesorem s označením Intel i3, který nabízí podporu VT-x, která je využívána pro hardwarovou virtualizaci a značně urychluje výkon při práci ve virtuálním prostředí s procesory Intel.

Obrázek 11 Notebook HP ProBook 4720s [26]

Po stránce softwarové jsem použil program Oracle VM Virtualbox, který je spuštěn pod operačním systémem Windows 7. Tento program představuje multiplatformní virtualizační

nástroj, který je distribuován pro systémy Linux, MacOS a Windows. V současné době je vyvíjen společností Oracle Corporation, a to i pro české jazykové prostředí.

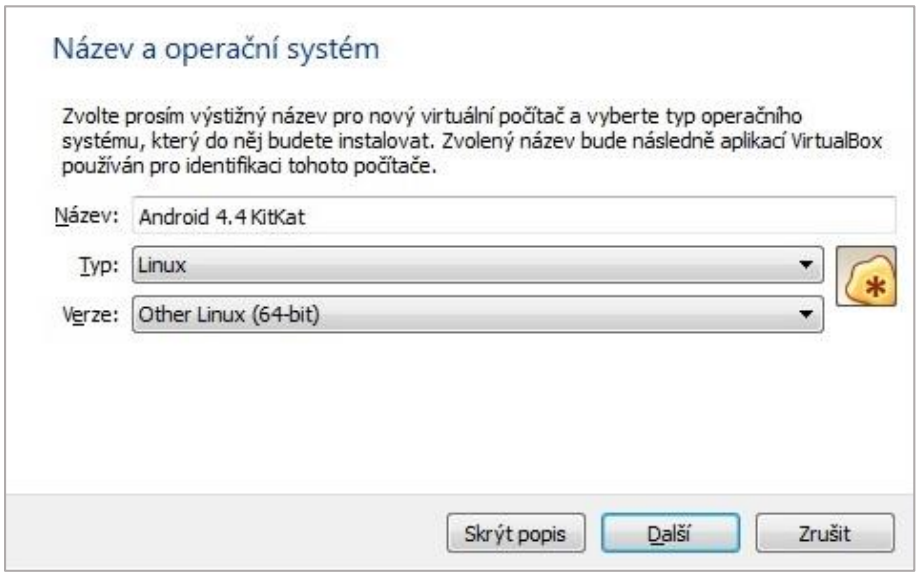
Obrázek 12 Virtualizační software VirtualBox

Prvním krokem pro instalaci programu VirtualBox je stáhnutí instalačního balíčku ze stránek výrobce (https://www.virtualbox.org/wiki/Downloads). Jedná se o základní instalační balíček pro platformu Windows. Po úspěšném nainstalování základního programu je nutné nainstalovat také rozšiřující balíček (Extension Pack), který je rovněž dostupný na stejných stránkách výrobce. Jakmile je vše připraveno, může se přistoupit k vlastní instalaci operačního systému Android.

5.1.1 Instalace OS Android

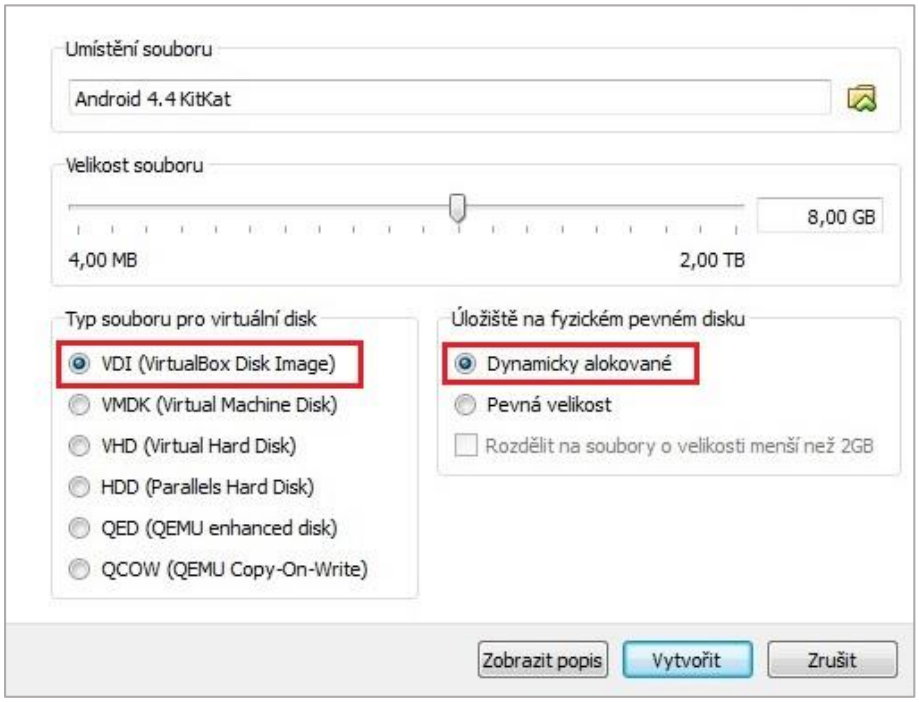
Pro instalaci operačního systému nejprve musíme stáhnout soubor (ISO). Tento soubor je dostupný na internetových stránkách (www.android-x86.org/download), a to vždy pro konkrétní verzi operačního systému. V našem případě budeme používat nejnovější verzi pod označením Android 4.4 KitKat určenou pro zařízení tablet.

Na hlavní obrazovce programu začneme tím, že vytvoříme nový virtuální stroj kliknutím na tlačítko *Nový*. Tím se začne instalační proces, během kterého je uživatel vyzván k zadání názvu virtuálního stroje a k výběru typu a verze operačního systému. Tento krok je zobrazen na obrázku 13.



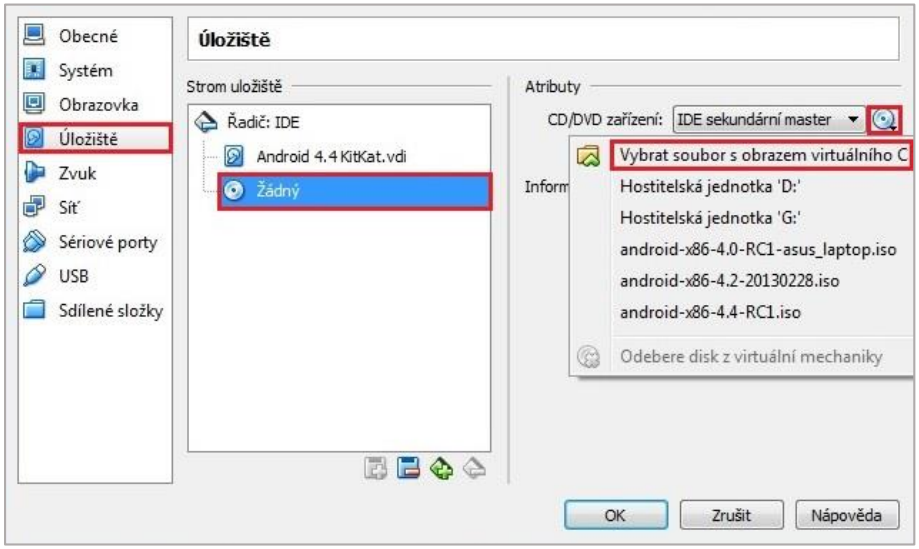
Obrázek 13 Vytvoření virtuálního počítače

Dalším krokem instalačního procesu je zadání velikosti paměti RAM (Random Access Memory), která bude alokována pro virtuální počítač. V této části si alokujeme takovou část paměti, aby nám zbylo dostatečné množství pro chod samotného systému Windows. Po zadání velikosti paměti bude uživatel vyzván k výběru virtuálního disku, který bude systém Android používat. V tomto případě zvolíme dynamický formát používaný VirtualBoxem VDI (VirtualBox Disk Image). Tento krok je spolu s příslušnými volbami zobrazen na obrázku 14.



Obrázek 14 Vytvoření virtuálního pevného disku

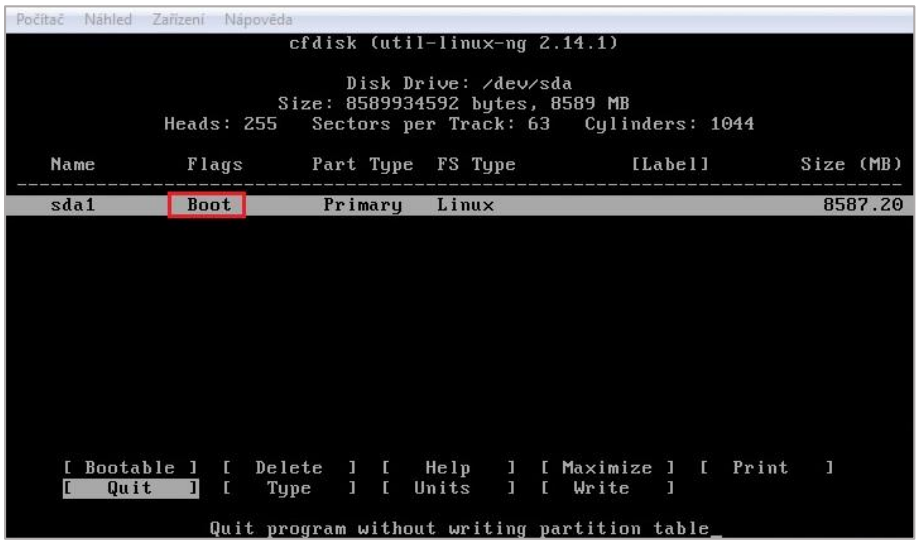
V tuto chvíli máme vytvořený nový virtuální stroj pro systém Android. Nyní musíme provést základní nastavení a potom tento systém nainstalovat prostřednictvím přiděleného souboru (ISO). Po kliknutí na ikonu *Nastavení* se spustí navigační okno, ve kterém zvolíme záložku *Úložiště*. V této záložce klikneme na ikonu mechaniky CD a v pravé části kliknutím na stejnou ikonu vybereme možnost *Vybrat soubor s obrazem virtuálního CD/DVD*. V následující nabídce vybereme soubor (ISO) pro systém Android. Tyto kroky jsou zobrazeny na následujícím obrázku.



Obrázek 15 Přiřazení virtuálního obrazu (ISO)

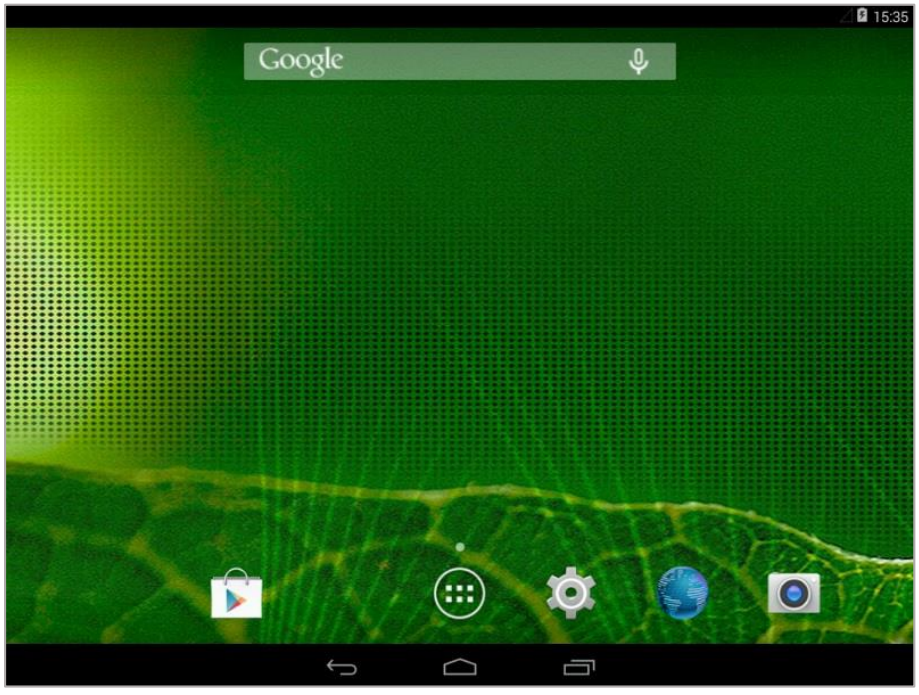
Nyní máme systém připravený na první spuštění, při kterém se spustí proces instalace na pevný disk počítače. Instalaci zahájíme kliknutím na ikonu *Spustit*.

Po zahájení procesu instalace jsme systémem vyzváni k výběru druhu spouštění, kdy zadáme volbu *Install Android-x86 to harddisk*. To nám zajistí, že systém se nainstaluje na pevný disk počítače, a proto nebude nutné při každém jeho spuštění opětovně zadávat výběr základních systémových předvoleb. Dalším krokem, který potvrdíme volbou *Create/Modify partitions*, bude vytvoření bootovacího oddílu. Toto nastavení provedeme tak, jak je zobrazeno na obrázku 16, a ukončíme volbou *Quit*.



Obrázek 16 Obrazovka nastavení systémového oddílu

Po úspěšném nastavení systémového oddílu tento oddíl vybereme, zvolíme pro něj systémový formát *ext3* a celý proces přípravy potvrdíme oprávněním umožňujícím zápis i čtení z disku. V tuto chvíli je systém nainstalovaný na pevném disku a stačí jej pouze spustit. V případě prvního spuštění systému je nutné zadat základní systémové volby, jako je výběr jazyka, nastavení sítě Wifi a nastavení uživatelského účtu společnosti Google. Volba nastavení tohoto účtu je přitom velmi důležitá, protože prostřednictvím účtu Google se lze připojit k internetovému obchodu Google Play, ze kterého bude nutné nainstalovat některé aplikace pro další část práce.



Obrázek 17 Úvodní obrazovka OS Android

5.2 Detekce škodlivé aplikace

V této části si vyzkoušíme za použití škodlivé aplikace, jak systém Android reaguje v případě, kdy se uživatel pokusí takovou aplikaci nainstalovat do svého zařízení. Bude zde vyzkoušena především nová funkce Verify apps ověřující škodlivost aplikace ještě před její instalací. Tato funkce je doplňkem pro novou službu Google Bouncer, která má za úkol automaticky testovat všechny aplikace, které vývojáři umístí do internetového obchodu Google Play.

Jako první krok testu musíme nejdříve do zařízení nainstalovat některou z aplikací pro správu souborů. To nám zajistí, že instalaci škodlivé aplikace bude možné provést přímo z úložiště v zařízení prostřednictvím instalačního balíčku (APK). Pro tento krok jsem vybral aplikaci *Správce souborů (File Manager)*, která je zdarma dostupná v obchodě Google Play.

Jako další krok stáhneme do zařízení instalační balíček vybrané škodlivé aplikace. V tomto případě jsem vybral aplikaci ve formě hry, která má název *Flappy Bird* a v případě, že bude v zařízení spuštěna, umožňuje odesílat prémiové zprávy SMS nebo odesílat citlivá data ze zařízení na vybrané IP adresy.

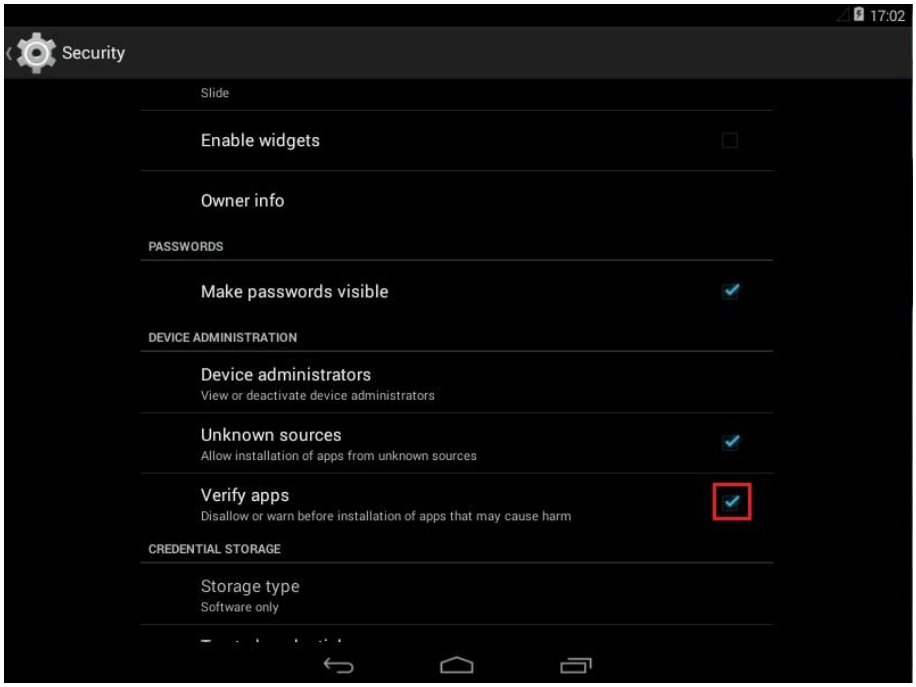
V případě, že chceme do zařízení nahrát některou z aplikací, můžeme tak učinit těmito možnými způsoby:

- **prostřednictvím oficiálního obchodu** (v tomto případě lze aplikaci do zařízení přímo instalovat a nikoli pouze nahrát na úložiště. Tento způsob pořízení aplikace by měl být pro uživatele bezpečný. Ve skutečnosti tomu tak je pouze u systémů iOS a Windows Phone, kdy jsou všechny aplikace kontrolovány na škodlivý kód ještě před uvedením na trh. U systému Android je takto aplikace zkontrolována a vyřazena z obchodu až v případě, kdy jsou skutečnosti o její škodlivosti oznámeny společnosti Google),
- **prostřednictvím bezdrátového rozhraní Bluetooth** (tento případ neumožňuje pro uživatele provést žádnou kontrolu při samotném stáhnutí aplikace a uživatel by si tak měl být jist, že je sdílené zařízení důvěryhodné),
- **prostřednictvím webových stránek** (takto lze do zařízení aplikaci stáhnout za pomocí webového prohlížeče. Ani v tomto případě však nemá uživatel žádnou jistotu, zda se aplikace nebude chovat škodlivě. Proto musí i zde zvážit, zda je tento

zdroj věrohodný. U systému iOS a Windows Phone není možné aplikaci v zařízení takto instalovat, pokud není distributorovi vydán patřičný certifikát),

- **prostřednictvím přílohy obsažené v e-mailové zprávě** (v případě, že uživatel používá e-mailovou službu, která je jejím poskytovatelem kontrolována na přítomnost malwaru antivirovým programem, tak je o případné detekci malwaru informován a škodlivý soubor není přenesen. Pozor si však musí dát v případě, kdy je přenášený soubor zkomprimován a zabezpečen heslem. Do takového souboru nemá antivirový program přístup a ten je prostřednictvím poštovního klienta přenesen do zařízení),
- **prostřednictvím externího úložiště MicroSD** (u externích úložišť systém neprovádí žádnou kontrolu, zda neobsahují škodlivé aplikace a uživatel by si měl dodatečně ověřit, zda je obsah na těchto kartách nezávadný. To lze v případě systému Android udělat např. za pomocí antivirových produktů třetích stran. U zařízení Apple nejsou externí úložiště výrobcem podporována).

Jakmile je škodlivá aplikace nahrána do úložiště zařízení, můžeme se ji pokusit nainstalovat a ověřit, jak bude funkce systému Verify apps v takovém případě reagovat. Ještě než spustíme instalaci, musíme se přesvědčit, že je tato funkce aktivována. To zjistíme kliknutím na ikonu *Settings* (*Nastavení*), kde v záložce *Security* (*Zabezpečení*) musí být aktivována funkce Verify apps (Ověření aplikací) tak, jak je zobrazeno na následujícím obrázku.



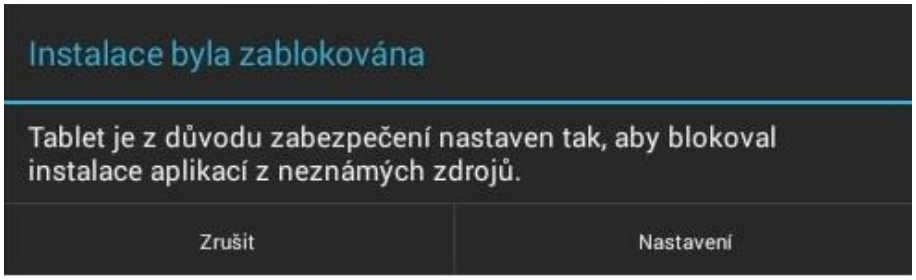
Obrázek 18 Aktivace funkce Verify apps

Nyní spustíme aplikaci Správce souborů, kterou jsme si do zařízení nainstalovali jako první. Zde máme zobrazenou strukturu adresářů celého operačního systému. Po vstupu do adresáře *Download* v něm nalezneme škodlivou aplikace *Flappy Bird*, kterou jsme si do zařízení nahráli, v tomto případě prostřednictvím webového prohlížeče. Aplikace je zde uložena ve formě instalačního balíčku (APK). Spuštěním tohoto balíčku se automaticky aktivuje instalační proces, který je v případě systému Android provázen dvěma bezpečnostními prvky:

- Unknown sources (Neznámé zdroje)
- Verify apps (Ověření aplikací)

Unknown sources (Neznámé zdroje)

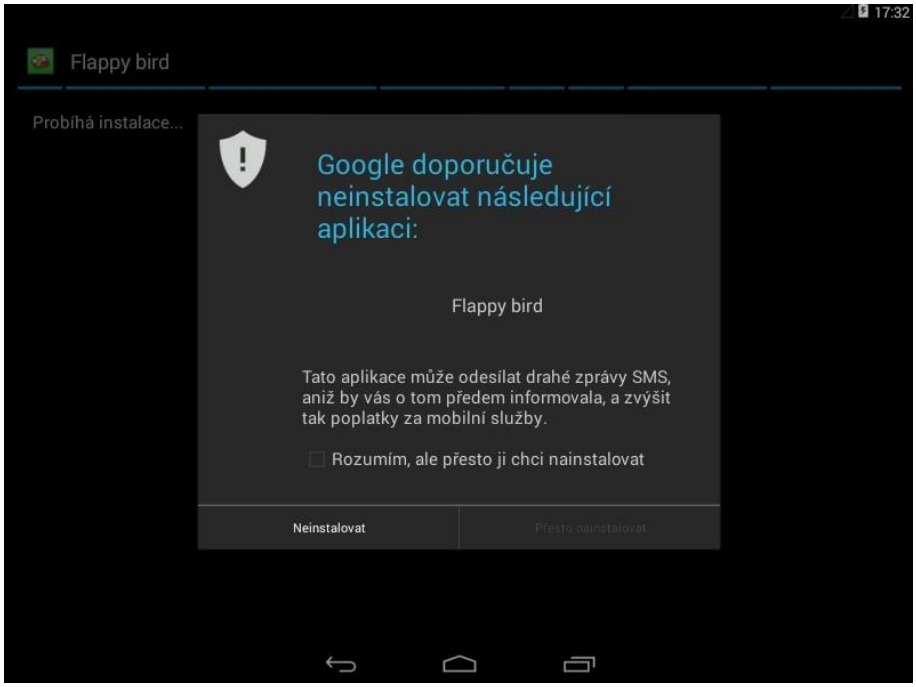
V tomto případě se jedná o možnost povolení instalace aplikací z neznámých zdrojů. V praxi to znamená, že uživatel v případě, kdy má tuto volbu povolenou, může aplikace nainstalovat i z jiného zdroje, nežli je oficiální obchod Google Play. V našem případě musí být tato volba aktivována (viz. Obrázek 18). V případě, že tak neučiníme, bude instalace systémem zablokována a my budeme na tuto blokaci upozorněni tak, jak je zobrazeno na obrázku 19.



Obrázek 19 Varování pro instalaci z neznámých zdrojů

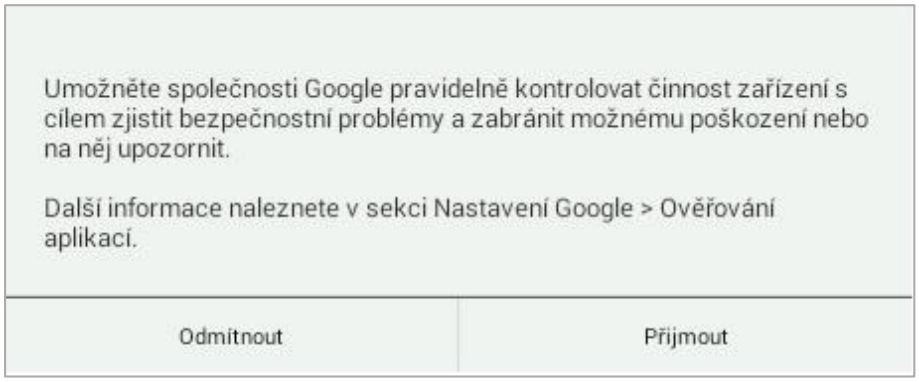
Verify apps (Ověření aplikací)

Google play zpřístupňuje od verze Android 4.2 funkci Verify apps, která uživatele při instalaci upozorní na potenciálně nebezpečnou aplikaci. V případě, že je tato funkce aktivována (viz. Obrázek 18), odešlou se ze zařízení na server Google informace, které tuto aplikaci ověří. V případě, že se jedná o škodlivou aplikaci, tak se zobrazí dvě možná upozornění. Prvním upozorněním je pouze doporučení (viz. Obrázek 20), abychom tuto aplikaci neinstalovali. Jedná se například o detekci podezřelých oprávnění. V tomto případě však lze, zaškrtnutím políčka *Přesto nainstalovat,* aplikaci do zařízení nainstalovat. Ve druhém případě se zobrazí hlášení, které instalaci rovnou zablokuje, kdy uživatel toto hlášení nemůže nijak ovlivnit. V tomto případě se může jednat o podezření například na exploit.



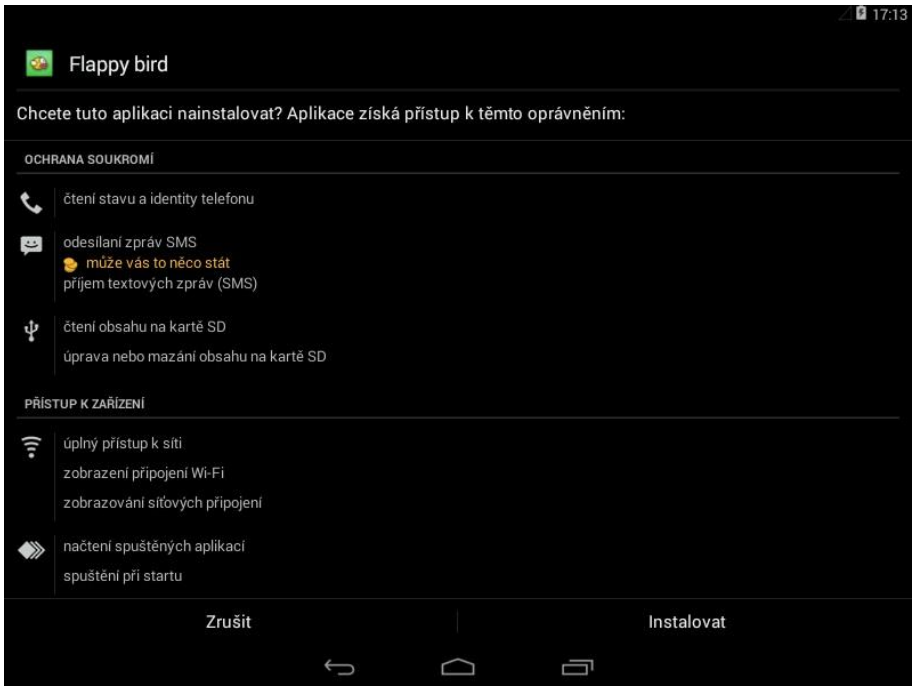
Obrázek 20 Upozornění funkce Verify apps

Funkce Verify apps bývá ve výchozím nastavení při instalaci operačního systému aktivována. To však neznamená, že se hned budou ze zařízení odesílat ověřující informace. V tomto případě musíme odesílání těchto dat potvrdit při první instalaci některé z aplikací (viz. Obrázek 21).



Obrázek 21 Upozornění na odesílání informací pro ověření

Dalším krokem v instalačním procesu bude zobrazení seznamu oprávnění, která budou aplikaci po nainstalování zpřístupněna. Jedná se o oprávnění, která zadává vývojář každé aplikace již při její tvorbě. V tomto případě nelze souhlasit pouze s některými oprávněními a uživatel je nucen odsouhlasit všechna oprávnění bez výjimky. V případě naší testující aplikace *Flappy Bird* jsou tato oprávnění zobrazena na následujícím obrázku.



Obrázek 22 Přístupová oprávnění aplikace Flappy Bird

Kontrola přístupových oprávnění každé aplikace je jednou z nejdůležitějších činností instalačního procesu, kterou kontroluje každý uživatel sám. Této kontrole je nezbytné věnovat zvýšenou pozornost, kdy je důležité se zaměřit především na to, zda jednotlivá oprávnění jsou pro aplikaci opravdu důležitá a nezbytná. V případě naší aplikace je na předchozím obrázku vidět, že aplikace vyžaduje přístup k oprávněním pro odesílání a příjem zpráv SMS. Navíc je zde uvedeno, že tyto zprávy mohou být pro uživatele zpoplatněny. Již v tomto případě by si měl uživatel uvědomit, proč by aplikace kategorie hry měla vyžadovat přístup k tomuto oprávnění. Toto je první varování, že by se mohlo jednat o škodlivou aplikaci. Ne však každá aplikace vyžadující oprávnění pro odesílání zpoplatněných textových zpráv musí být škodlivá. Může se také jednat o aplikaci, kterou uživatel využívá např. pro zakoupení jízdenky městské hromadné dopravy. V tomto případě jsou tato oprávnění zcela v pořádku a pro uživatele nepředstavuje taková aplikace žádné nebezpeční.

Jak lze tedy ověřit, zda je aplikace závadná či nikoliv? V případě, kdy si uživatel není zcela jist, zda aplikace představuje nebezpečí, může si prostřednictvím statické a dynamické analýzy nezávadnost aplikace ověřit. Tyto postupy jsou vysvětleny v následující kapitole Identifikace malwaru.

5.3 Identifikace malwaru

V této kapitole si řekneme, jak lze ověřit, zda aplikace, kterou se chystáme nainstalovat do svého zařízení, představuje, či nepředstavuje nebezpečí. K tomuto účelu využijeme některých webových služeb, které se touto tematikou zabývají a výsledky svých analýz poskytují zcela zdarma. Těchto služeb je v dnešní době celá řada a pro detekci a identifikaci malwaru využívají techniky statické a dynamické analýzy.

Statická analýza

Statická analýza bývá prvním krokem k identifikaci malwaru. Během této analýzy je soubor podroben řadě procesů, mezi které patří např. porovnání údajů v databázi o virech nebo kontrola integrity prostřednictvím hashovacích funkcí. V těchto případech se nemusíme obávat žádného nebezpečí, jelikož zdrojový kód souboru není ve skutečnosti spuštěn a systém tak nemůže být ohrožen.

Dynamická analýza

Dynamická analýza je méně bezpečnou technikou v identifikaci malwaru z důvodu, že zdrojový kód programu je analyzován za běhu, a tudíž může do jisté míry představovat nebezpečí. Z tohoto důvodu je pro tuto analýzu využíváno nejrůznějších emulátorů.

V následující části kapitoly si ukážeme obě tyto techniky prostřednictvím vybraných webových služeb. V prvním případě se jedná o webovou službu *VirusTotal* dostupnou na adrese (https://www.virustotal.com). Zde si na úvodní obrazovce můžeme vybrat ze tří možností. Jedná se o možnost kontroly celého souboru, možnost kontroly internetové adresy nebo vyhledání informací dle zadaných hodnot. V našem případě využijeme naší testovací aplikace *Flappy Bird*, se kterou jsme v předchozí kapitole pracovali. Vybereme tedy tento instalační balíček ve formě souboru (APK) a zahájíme proces analýzy potvrzením tlačítka *Scan it!*. Tento krok je zobrazen na následujícím obrázku.

Obrázek 23 Úvodní obrazovka služby VirusTotal

Po spuštění procesu analýzy stačí jen zvolit, zda chceme zobrazit již provedenou analýzu nebo chceme provést analýzu zcela novou.

Výhodou služby *VirusTotal* je, že analýza neprobíhá pouze prostřednictvím jednoho antiviru, ale je prováděna na několika antivirech současně. To je důležité, protože v případě, kdy některý z antivirů neoznačí soubor aplikace za škodlivý, může tak učinit jiný, čímž se značně zvyšuje pravděpodobnost bezchybné detekce.

Po ukončení procesu analýzy se na obrazovce zobrazí výsledný identifikační údaj pro identifikaci malwaru, který se nachází zvlášť u každého antiviru. Tento údaj je zobrazen

podle základního schématu dle organizace CORA[47] (Computer Antivirus Research Organization) a vychází z následujícího obecného formátu, jehož pořadí se může lišit dle jednotlivých antivirových produktů [19].

- <platform>.<family_name>.<group_name>.<variant>.<modifiers>

platform (platforma)

Tato informace určuje, pro který operační systém je hrozba určena. Jedná se o operační systémy, jako je např. Android, iOS, Windows Phone nebo jiné operační systémy pro desktopové počítače. Položka může obsahovat také programovací jazyky nebo formáty souborů.

family_name (jméno rodiny)

Znamená, do které rodiny můžeme malware zařadit. Jedná se o rodiny, jako jsou viry, červi, trojské koně, adware, spyware atd.

group_name (jméno skupiny)

Určuje, jakou hrozbu program pro operační systém představuje. V tomto případě myslíme určitou skupinu obdobných kódů v rámci některé z rodin.

variant (varianta)

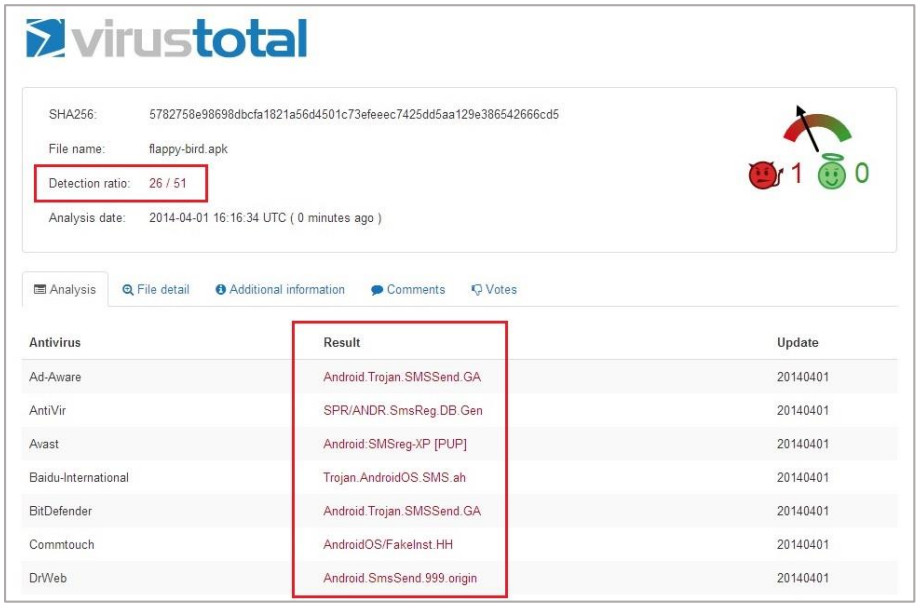
Varianta představuje číslo reprezentující infekční délku kódu a používá se pro určitou skupinu kódů, které jsou si velmi podobné a mají jednu a tutéž infekční délku. Dá se tedy říct, že prostřednictvím tohoto čísla lze rozlišit určitou podskupinu kódů v rámci jedné skupiny.

modifiers (modifikace)

V některých případech obsahuje syntaxe položku zvanou modifikace. To se týká především polymorfních virů, které jsou schopny skrýt se před antivirovými programy tak, že dokáží modifikovat svůj zdrojový kód například prostřednictvím komprese. Takový virus potom dokáže uniknout skenování a tváří se, jako by byl zcela novým virem. Položka modifikace tedy určuje, jaký skrývající mechanismus byl použit.

[47] CARO je označení pro neformální organizací sdružující skupinu jednotlivců, kteří se zabývají studiem a výzkumem počítačového malwaru. Skupina byla založena v roce 1990.

Nyní se podívejme na obrázku 24, jak v případě naší aplikace proběhla analýza u vybraných antivirových programů. Všimněme si u položky *Detection ratio*, že bylo pro analýzu použito celkem 51 antivirových programů, kdy aplikaci označilo za škodlivou 26 z těchto programů.

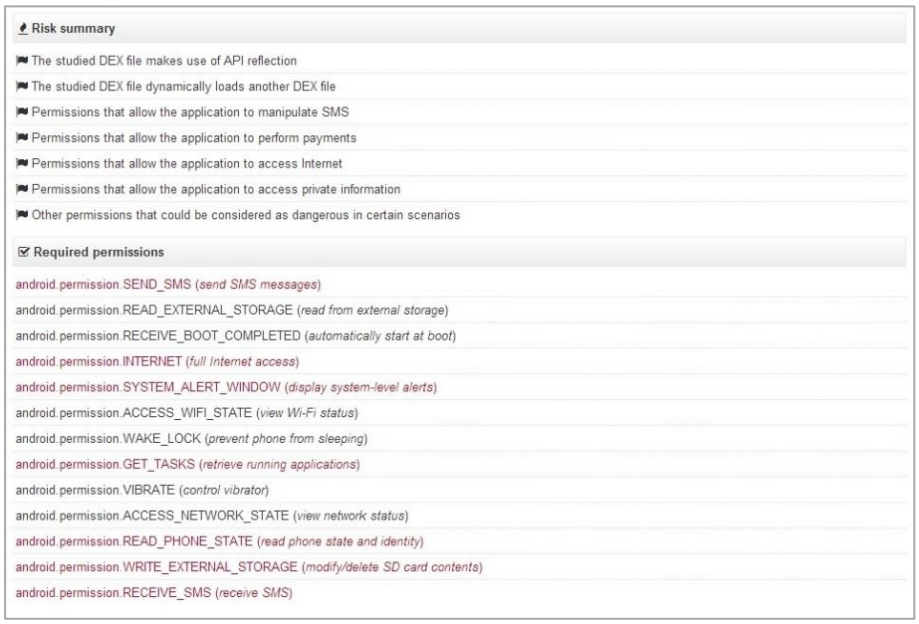


Obrázek 24 Výsledek analýzy služby VirusTotal

Uprostřed stejného obrázku je u každého antivirového programu zobrazena syntaxe škodlivého kódu. Téměř ve všech případech je vidět, že tato aplikace je za škodlivou identifikována především na základě možnosti odesílat a přijímat zprávy SMS. Ve skutečnosti však tato aplikace využívá i dalších oprávnění, která mohou být také označena za zdroj škodlivé aplikace. Všechna tato oprávnění se zobrazí po zvolení záložky *File detail* (viz. Obrázek 25).

5.3.1 Přístupová oprávnění API

Na obrázku 25 jsou v záložce *File detail* v horní části zobrazena rizika, která jsou dána přístupem k jednotlivým oprávněním, jenž má možnost aplikace využívat. Ve spodní části jsou vypsána tato oprávnění tak, jak jsou vývojářem deklarována v souboru (AndroidManifest.xml), který je obsažen v kořenovém adresáři instalačního balíčku (APK). Jedná se o stejná oprávnění, jaká jsou zobrazena uživateli při instalačním procesu.



Obrázek 25 Detailní analýza zpřístupněných oprávnění

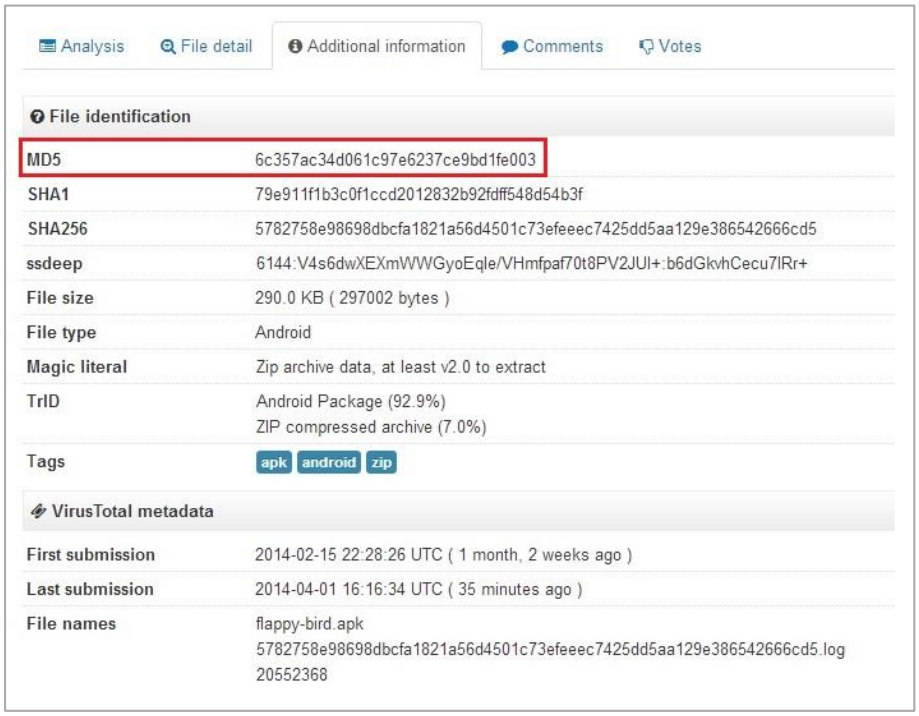
Červeným textem na přechozím obrázku jsou zvýrazněna oprávnění, jenž jsou analýzou vyhodnocena jako riziková a představují pro uživatele jisté nebezpečí. V případě testující aplikace jsou to tato oprávnění:

- *android.permission.SEND_SMS* (oprávnění umožňuje aplikaci odesílat zprávy SMS),
- *android.permission.READ_PHONE_STATE* (oprávnění umožňuje přístup pro čtení stavu a identity telefonu),
- *android.permission.SYSTEM_ALERT_WINDOW* (oprávnění umožňuje aplikaci otevření vyskakujícího okna nad úrovní všech ostatních aplikací. Toto oprávnění by mělo využívat jen velmi málo aplikací, jelikož je určeno pro přímou interakci systému s uživatelem),
- *android.permission.GET_TASKS* (umožňuje aplikaci získat informace o spuštěných úlohách),
- *android.permission.INTERNET* (umožňuje aplikaci plný přístup k internetu. Jedná se zejména o možnost otevřít síťové sockety),
- *android.permission.WRITE_EXTERNAL_STORAGE* (umožňuje zapisovat na externí úložiště),
- *android.permission.RECEIVE_SMS* (umožňuje aplikaci přijímat zprávy SMS).

Po analýze jednotlivých oprávnění je zřejmé, že aplikace využívá celou řadu oprávnění, dle kterých by mohla být vyhodnocena jako nebezpečná. Nejedná se tedy pouze o riziko

spojené s odesíláním prémiových zpráv, ale také o rizika odesílání důvěryhodných informací prostřednictvím internetového spojení.

Služba *VirusTotal* nám poskytuje další důležité informace, jako je typ souboru, velikost souboru, přesné datum provedené analýzy a ukazuje, zda nedošlo při prováděné analýze k úmyslnému předložení jiného souboru. Tato možnost se kontroluje prostřednictvím hodnoty hashovací funkce MD5 (Message Digest 5), SHA1 nebo SHA256. Je to zároveň jedna z hodnot, které se odesílají a kontrolují v rámci bezpečnostní funkce Verify apps, která byla prezentována v rámci této kapitoly. Tyto hodnoty jsou zobrazeny v záložce *Additional information* tak, jak ukazuje následující obrázek.

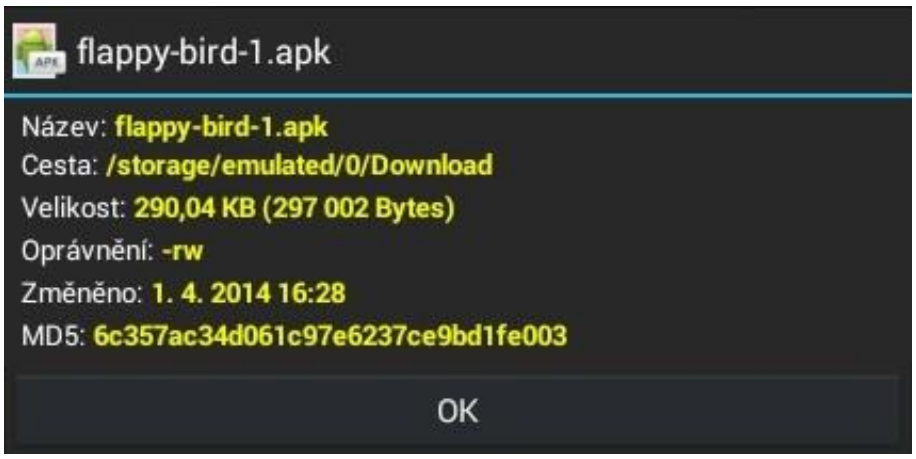


Obrázek 26 Kontrola hodnoty hashovacích funkcí

Pro kontrolu integrity souboru v našem případě postačí hodnota funkce MD5. Nyní je však nezbytné zjistit, zda je tato hodnota stejná jako v případě instalace naší aplikace. V tomto případě máme několik možností, jak se o tom přesvědčit. První možností je nainstalování programu, který je schopen tuto hodnotu vygenerovat. Programy jsou dostupné jak pro desktopové operační systémy, tak přímo v rámci aplikace v našem zařízení. V našem případě však využijeme opět aplikace *Správce souborů*, ve které si hodnotu MD5 zobrazíme a následně porovnáme s hodnotou vygenerovanou v rámci služby *VirusTotal*.

Nyní se vrátíme ke kroku, kdy jsme se chystali instalovat aplikaci *Flappy Bird* do zařízení a v aplikaci *Správce souborů* se přesuneme opět do složky *Download*. V tomto případě

však podržením tlačítka necháme zobrazit nabídkové okno, ve kterém vybereme volbu *Podrobnosti*. Tato volba nám zobrazí informační okno, kde je zobrazena hodnota MD5 (viz. Obrázek 27).

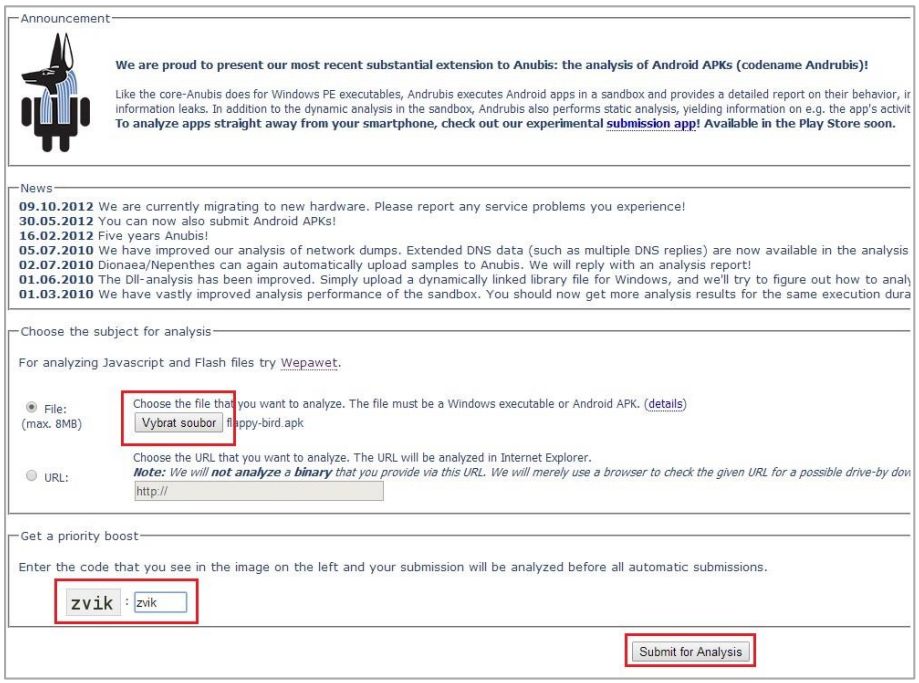


Obrázek 27 Zobrazení hodnoty MD5

Jak je vidět na předchozích obrázcích, tak obě hodnoty jsou naprosto totožné. To nám zaručuje, že naše aplikace je přesně ta, která byla testována prostřednictvím služby *VirusTotal*.

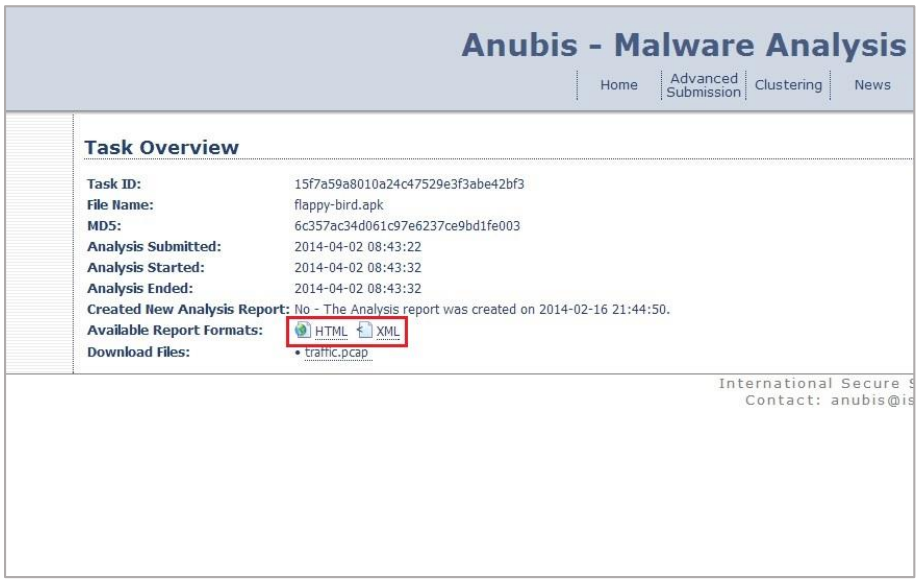
V další části práce pomocí webové služby *Anubis* provedeme dynamickou analýzu struktury binárního kódu a prostřednictvím této struktury určíme, která část kódu představuje pro uživatele nebezpečí. Opět k této činnosti využijeme aplikaci *Flappy Bird*.

Nejprve prostřednictvím webového prohlížeče zvolíme adresu (https://anubis.iseclab.org). Po zobrazení hlavní nabídky zvolíme soubor pro testování kliknutím na tlačítko *Vybrat soubor* a potom opíšeme vygenerovaný bezpečnostní kód. Jakmile tak učiníme, spustíme proces analýzy tlačítkem *Submit for Analysis* nacházející se ve spodní části obrazovky (viz. Obrázek 28).



Obrázek 28 Výběr souboru pro dynamickou analýzu

Jakmile se odešle instalační soubor (APK) na vzdálený server, bude provedena statická i dynamická analýza, která v některých případech může trvat několik minut (zpravidla 10-15 minut). To záleží, zda byl soubor již testován nebo se jedná o zcela novou analýzu. Jakmile bude proces ukončen, ihned se zobrazí obrazovka ukončení analýzy s nabídkou k nahlédnutí do dvou souborů (viz. Obrázek 29).



Obrázek 29 Ukončení analýzy aplikace

Prvním souborem je soubor typu HTML (HyperText Markup Language), který obsahuje veškeré informace rozdělené do kategorie statické a dynamické analýzy. V případě statické

analýzy jsou tyto informace obdobné jako v případě analýzy službou *VirusTotal* (hodnoty hashovacích funkcí, jednotlivá přístupová oprávnění a další informace). V našem případě se zde zaměříme na záložku *Sent SMS*, která je obsažena ve výpisu dynamické analýzy. Tato záložka, která je vidět na následujícím obrázku, obsahuje informaci o odeslání dvou prémiových zpráv SMS na předem zvolené číslo. Tato aktivita byla zaznamenána během běžícího procesu aplikace ve virtuálním prostředí.

Obrázek 30 Část výpisu dynamické analýzy

Druhým souborem ve výsledku analýzy je soubor typu (XML), který zobrazuje výstup kódu aplikace za jejího provozu. Na obrázku 31 je vidět, ve které části kódu byly zprávy SMS odeslány a na jaké číslo. Na stejném obrázku jsou také vidět podezřelé aktivity síťového provozu, kdy jsou data v šifrované podobě odesílána na předem zvolené IP adresy.

Obrázek 31 Výstup kódu dynamické analýzy

Závěrem této kapitoly se dá konstatovat, že v případě systému Android by si měl uživatel dávat pozor, jaké aplikace do svého zařízení instaluje a zda tyto aplikace pochází z důvěryhodných zdrojů. Opatrnost je na místě i v případě verze Android 4.2 nebo vyšší, která sice obsahuje funkci Verify apps, ale jak se již několikrát potvrdilo, není tato kontrola v současné době dostatečně spolehlivá.

5.4 Bezpečný spouštěcí proces OS iOS

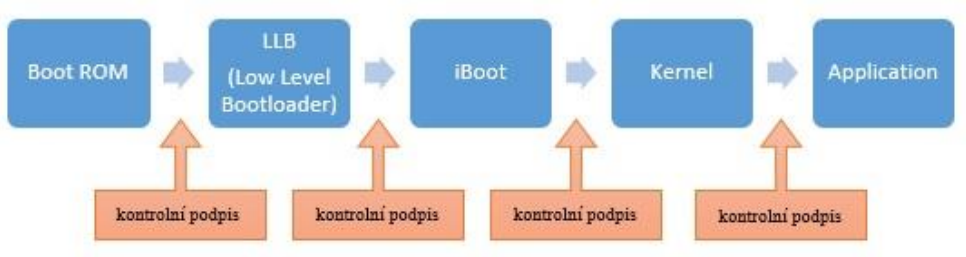
V případě systémů Windows Phone 8 a iOS je ochrana před malwarem dána dvěma základními způsoby. Tím prvním způsobem je, že každá aplikace je ještě před svým zveřejněním kontrolována týmy Microsoft a Apple. Ti ji zpřístupní prostřednictvím svých oficiálních obchodů až v případě, kdy takovou aplikaci shledají jako nezávadnou. Navíc tyto operační systémy neumožňují nainstalovat žádnou aplikaci, která nepochází z oficiálního obchodu a ani neumožňují provádět jakékoliv zásahy do běhu hlavního systému (např. formou systémových utilit). Výjimkou je pouze možnost distribuce a instalace aplikací v prostředí organizace. V tomto případě, je však nutné mít ověřený certifikát pro danou organizaci, který je vydán společností Microsoft nebo Apple, a to na základě splnění přísných registračních požadavků. Druhým způsobem ochrany systému je ochrana za pomoci bezpečného spouštěcího procesu. V tomto případě je operační systém při zapnutí zařízení spouštěn tak, že každá jeho část je před načtením nejprve kryptograficky ověřena, čímž je maximálně zajištěna integrita systému.

V poslední části této kapitoly si popíšeme princip tohoto spouštěcího procesu v případě systému iOS.

Spouštěcí řetězec operačního systému iOS se dělí do těchto pěti částí [21]:

- Boot ROM (Read Only Memory)
- LLB (Low Level Boot loader)
- iBoot
- Kernel
- Application

Celá sekvence spouštěcího řetězce je zobrazena na následujícím obrázku.



Obrázek 32 Spouštěcí řetězec systému iOS

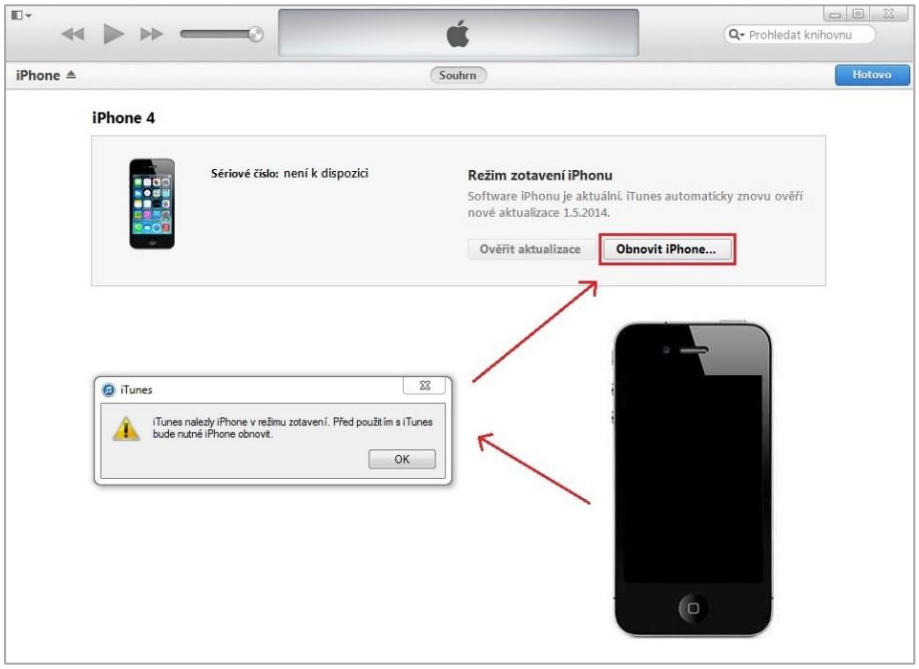
Boot ROM (Read Only Memory)

Boot ROM bývá uložen v části paměti umožňující pouze čtení. Tato paměť se nachází v systémovém čipu zařízení a je vytvořena již během jeho výroby. Toto opatření eliminuje riziko vyplývající s její neoprávněnou manipulací a zajišťuje vysokou integritu při spuštění bootovacího řetězce. V paměti je obsažena kořenová certifikační autorita Apple, která se používá k ověření podpisu každé části spouštěcího řetězce [21] [22].

LLB (Low Level Boot loader)

LLB je prvním krokem spouštěcího řetězce, který vyžaduje ověření podpisu prostřednictvím Boot ROM. V tomto případě se kryptograficky pomocí identifikátoru GID ověří, zda je možné zahájit proces iBoot a tím spustit firmware na konkrétní zařízení. Tento klíč GID je přitom zakotven v hardwaru zařízení a bývá pro určité typy zařízení stejný. V případě že, Boot ROM není schopen ověřit a načíst LLB, je zařízení uvedeno do DFU módu, kde prostřednictvím programu iTunes dojde k uvedení přístroje do továrního nastavení [15] [21] [22]. V tomto okamžiku lze sice propojit zařízení s iTunes, ale nelze

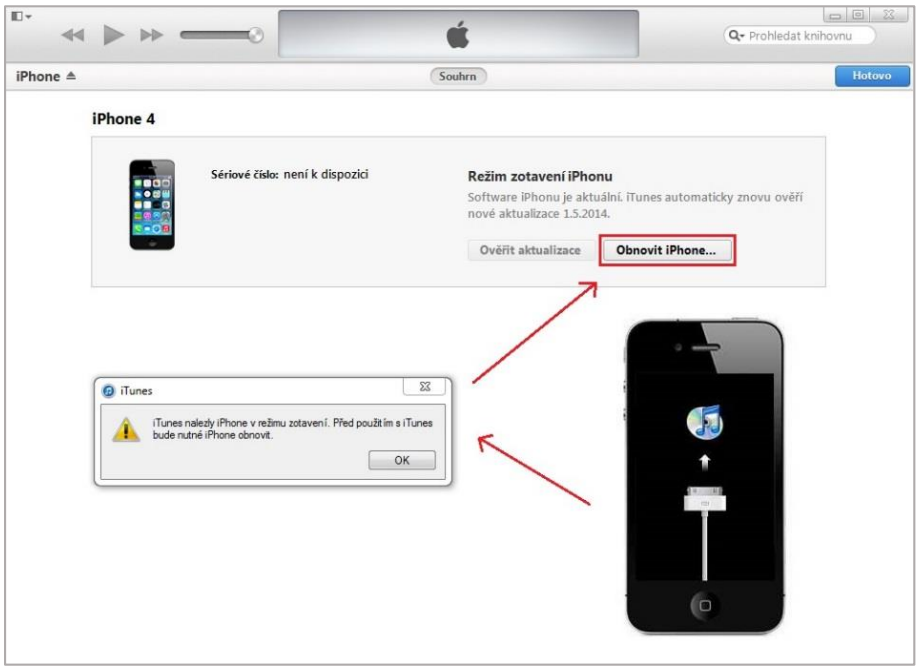
načíst operační systém nebo zavaděč. V tu chvíli se na displeji zařízení zobrazí černá obrazovka a zařízení je tak připraveno na změnu firmwaru (viz. Obrázek 33).



Obrázek 33 DFU mód systému iOS

iBoot

iBoot představuje druhý krok spouštěcího řetězce, který zajišťuje, aby během bootovacího procesu nedošlo k manipulaci softwaru na nejnižší úrovni bezpečnostní architektury. Proces zároveň disponuje podporou interaktivního USB nebo sériového rozhraní, které je využívané pro účely obnovy v případě selhání spouštěcího řetězce. Jakmile je během spouštěcího procesu tento krok kryptograficky ověřen, dojde k načtení jádra systému. V případě, že proces ověření selže, je zařízení uvedeno do tzv. Recovery mode (Režimu obnovy), kdy je nutné prostřednictvím programu iTunes zahájit proces obnovy (viz. Obrázek 34) [15] [21] [22]. V tomto případě však nelze do zařízení nahrát jiný firmware, ale pouze současně podporovaný uvést do továrního nastavení.



Obrázek 34 Mód obnovy systému iOS

Kernel

Po spuštění jádra se nejprve připojí systémový oddíl, který je určen pouze pro čtení. Tím je zajištěno, že v případě nuceného vypnutí zařízení zůstanou všechny systémové složky neporušeny. Dalším krokem je připojení oddílu s oprávněním k zápisu, na který se ukládají uživatelská data a aplikace. Ve všech případech jádro ověřuje podpis každého binárního kódu, který má být spuštěn. To zaručuje, že v systému iOS nemůže být spuštěn žádný binární kód ani aplikace, která není kryptograficky podepsána [21] [22].

Aplikace

V této části spouštěcího procesu je již načten systém a mohou být spouštěny jednotlivé aplikace. Ty se spouští vždy v ochranném módu Sandbox.

6 MOŽNOSTI ZABEZPEČENÍ MOBILNÍCH ZAŘÍZENÍ

V této kapitole se zaměříme na snížení rizik vyplývajících z problematiky ochrany mobilních zařízení. Budou zde ukázány některé postupy a rady, které mohou minimalizovat škody způsobené ztrátou nebo odcizením zařízení. V poslední části kapitoly se zaměříme na bezpečnost v podnikovém prostředí, kde si názorně ukážeme na systému iOS, jak lze chránit důvěrné informace v rámci organizace.

6.1 Obecné zásady bezpečnosti

Obecné zásady bezpečnosti se zabývají především problematikou využití bezpečnostních prvků a opatření, které poskytují samotná zařízení. Je zde kladen důraz také na uživatele, zda tyto prvky využívají či nikoli.

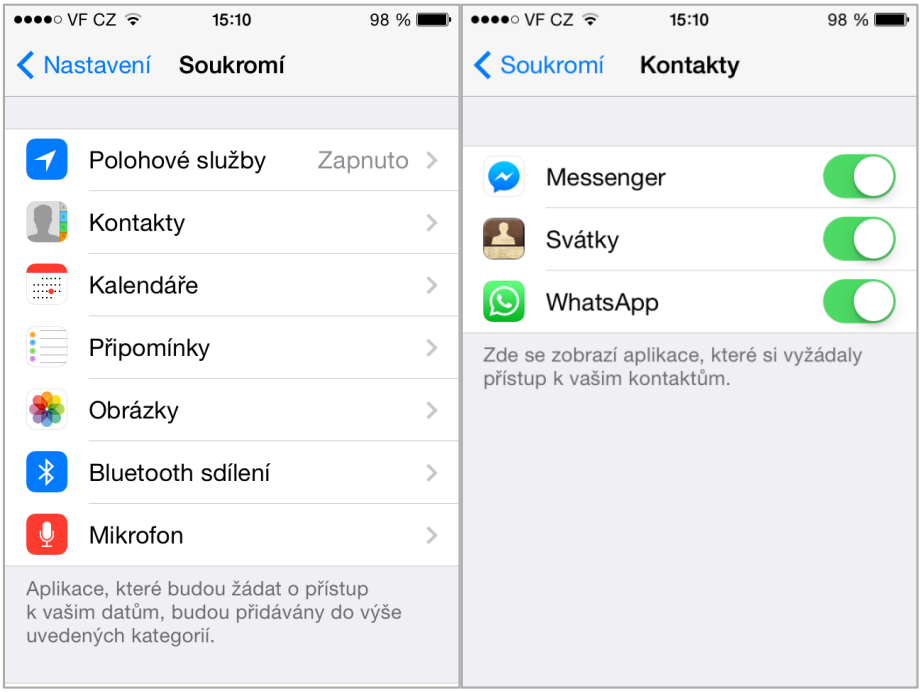
Mezi důležité bezpečnostní prvky a opatření patří:

- Uživatelská rozvážnost
- Kódový zámek
- Šifrování obsahu
- Zálohování dat
- Vzdálený přístup
- Bezpečné využívání Bluetooth a Wifi
- Aktualizace

6.1.1 Uživatelská rozvážnost

Ve všech případech výše uvedených operačních systémů je velmi důležité, aby si každý uživatel rozmyslel dopředu, na co zařízení bude potřebovat a co do něj bude instalovat. Tato problematika se o to více týká systému Android, který je na rozdíl od systému iOS a Windows Phone 8 zcela otevřenou platformou, kde může mít uživatel nad systémem neomezenou kontrolu a může tak třeba i nevědomě deaktivovat z provozu nejdůležitější bezpečnostní prvky *Verify apps* nebo *Unknown sources*. U tohoto systému je také velmi důležité kontrolovat přístupová oprávnění aplikací při jejich instalačním procesu a zvážit, zda všechna tato oprávnění jsou pro chod aplikace důležitá a nezbytná. Dobrým pomocníkem před instalací může být nahlédnutí do hodnocení jednotlivých aplikací včetně příspěvků v komentářích ostatních uživatelů. V případě systémů iOS a Windows Phone 8 jsou všechny aplikace kontrolovány ještě předtím, než jsou zveřejněny v oficiálním

obchodě a v tomto případě by nemělo uživatelům hrozit žádné nebezpečí. V praxi tomu tak však zcela vždy není, a proto je i zde obezřetnost ze strany uživatele zcela na místě. Na následujícím obrázku jsou zobrazena přístupová oprávnění systému iOS 7, kdy lze povolit nebo zablokovat jednotlivá oprávnění zvlášť pro každou aplikaci.

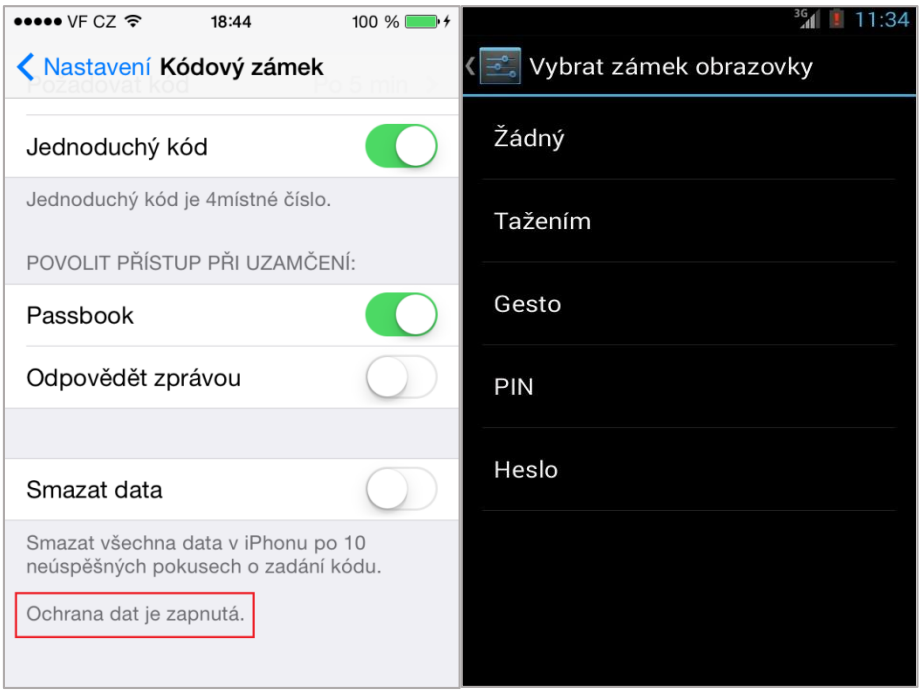


Obrázek 35 Zobrazení oprávnění systému iOS 7

6.1.2 Kódový zámek

Nastavení přístupového hesla by mělo být základním bezpečnostním prvkem každého zařízení. Ne však každý uživatel přístupového hesla využívá, a to z důvodu neustálého a obtěžujícího zadávání kódu v případě častého odemykání zařízení. U některých systémů, kde to dovoluje hardwarová technologie zařízení, je zámek obrazovky rozšířen o tzv. *Kódový zámek* (např. iOS). U takových zařízení by si měli uživatelé uvědomit, že právě nastavení kódového zámku umožňuje šifrování celého systémového úložiště. Při následné ztrátě nebo odcizení zařízení, nemůže útočník z takového zařízení odcizit žádné informace. V dalším případě by mělo každé zařízení opatřené přístupovým heslem útočníka alespoň na minimální dobu zpomalit, než se uživatel pokusí lokalizovat své zařízení nebo než z takového zařízení stihne vymazat veškerý obsah. Při volbě hesla je z hlediska bezpečnosti lepší volit alfanumerické zadání než zadání prostřednictvím čtyřmístného číselného kódu, a to z důvodu možného použití útoku hrubou silou. Další výhodou může

být možnost vymazání obsahu ze zařízení po několika neúspěšných pokusech o zadání hesla jako je tomu v případě systému iOS. Na obrázku 36 je v levé části zobrazeno nastavení kódového zámku pro systém iOS 7. Zde je možné si všimnout, že ochrana dat je v případě aktivace kódového zámku zapnutá. V případě, kdy se kódový zámek deaktivuje, se tato ochrana vypne a data jsou nechráněna. V pravé části stejného obrázku je zobrazeno nastavení klasického zámku obrazovky systému Android 4.3.

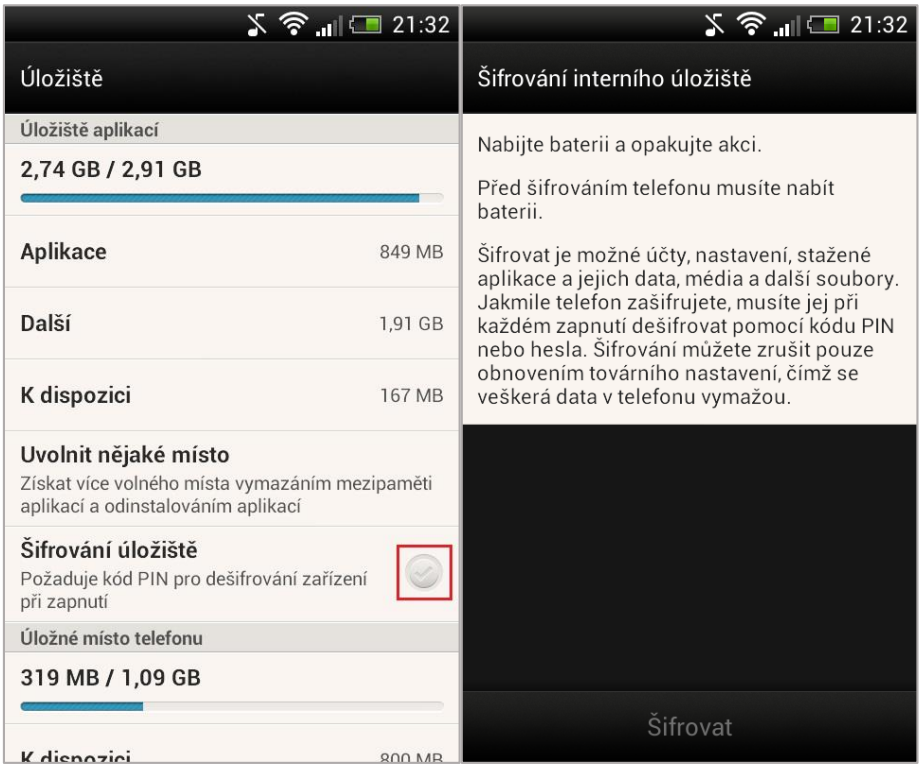


Obrázek 36 Klasický a kódový zámek zařízení

6.1.3 Šifrování dat

V případě, kdy se uživatel obává o zneužití svých dat při ztrátě nebo odcizení zařízení, měl by využít možnosti šifrování obsahu. Jak již bylo napsáno, systém iOS umožňuje šifrování dat na hardwarové úrovni. Toto šifrování se přitom aktivuje automaticky, jakmile uživatel aktivuje kódový zámek (viz. Obrázek 36). Možnosti hardwarového šifrování se liší podle operačního systému a jednotlivých zařízení, která musí být fyzicky vybavena touto podporou. V případě zařízení, která nemají možnost hardwarového šifrování, ale jejich operační systém šifrování umožňuje, lze šifrování aktivovat v nastavení systému nebo prostřednictvím aplikací třetích stran. Na obrázku 37 je zobrazena možnost šifrování interního úložiště systému Android 4.1.1, kterou lze aktivovat v nastavení systému v záložce *Úložiště*. V tomto případě je nutné nastavit kód PIN (Personal Identification

Number) nebo heslo, kterým se po každém zapnutí zařízení obsah úložiště dešifruje. Novější systémy Android umožňují také šifrování externích úložišť.



Obrázek 37 Šifrování úložiště systému Android

6.1.4 Zálohování dat

Uživatelé, kteří mají v zařízení uložena důležitá data a informace, popřípadě uživatelé, kteří využívají možnost vymazání obsahu zařízení na dálku nebo vymazání obsahu po několika neúspěšných pokusech, by měli svá zařízení pravidelně zálohovat. To jim zaručí, že při případné ztrátě nebo odcizení zařízení o svá data zcela nepřijdou a ty bude možné později obnovit v novém zařízení prostřednictvím počítače. Tyto zálohy lze provádět buďto prostřednictvím programů třetích stran, nebo prostřednictvím synchronizačních programů jednotlivých výrobců zařízení, které bývají poskytovány zpravidla zdarma. U některých poskytovatelů je navíc umožněno pro vyšší bezpečnost tyto zálohy šifrovat.

6.1.5 Vzdálený přístup

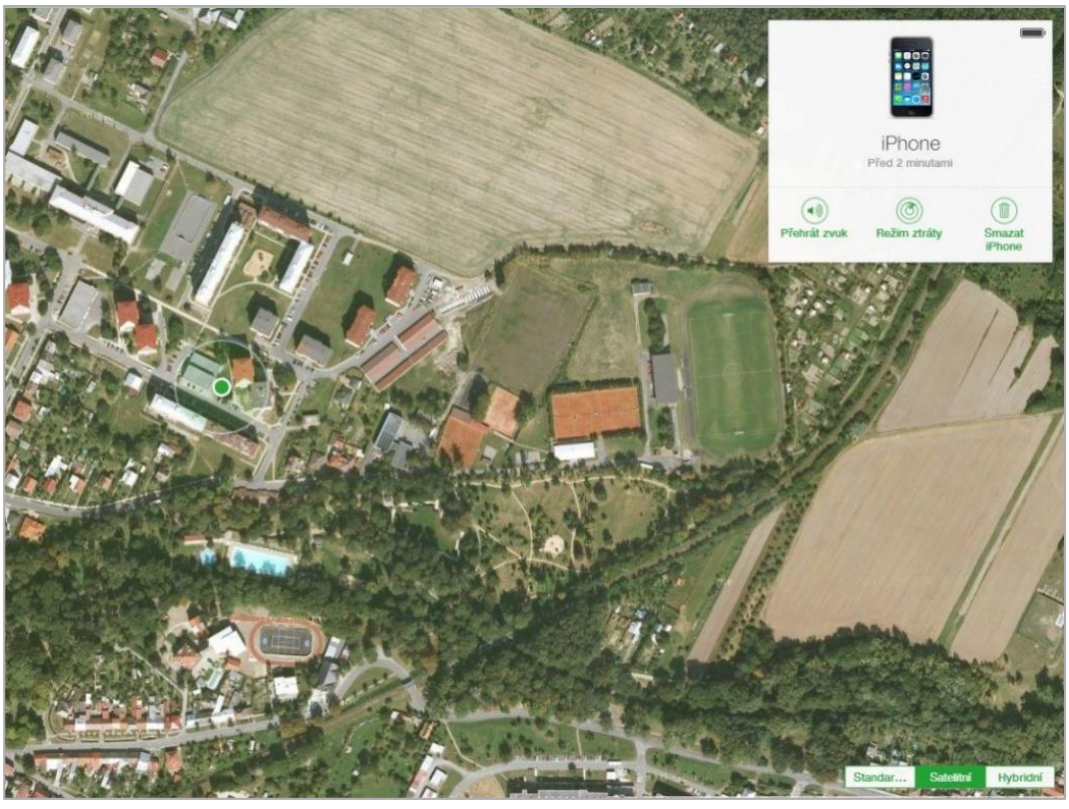
Vzdálený přístup k mobilním zařízením je dán prostřednictvím cloudových služeb jednotlivých výrobců. Ti prostřednictvím této služby umožňují uživatelům využívat řadu funkcí, mezi které patří především:

- Vzdálené vymazání zařízení

- Lokalizace zařízení na dálku
- Synchronizace a sdílení dat mezi více zařízeními (jedná se např. o fotografie, kontakty, kalendáře, poznámky, připomínky atd.)
- Zaslání zprávy nebo spuštění varovného signálu na ztraceném nebo odcizeném zařízení
- Zálohování zařízení na cloudové úložiště
- Správa e-mailových účtů
- Využívání vybraných webových aplikací prostřednictvím webového prohlížeče

Jedny z nejdůležitějších možností cloudových služeb z hlediska bezpečnosti je možnost vzdálené lokalizace zařízení a v případě ztráty nebo odcizení vymazání veškerého obsahu. Na následujícím obrázku je prostřednictvím služby iCloud od společnosti Apple vidět lokalizaci zařízení iPhone 4 s využitím mapového podkladu. V tomto případě lze rovněž zvolit jednu ze tří možností:

- Přehrání zvuku
- Režim ztráty
- Smazat iPhone



Obrázek 38 Lokalizace zařízení pomocí služby iCloud

První možností je vzdálené přehrání zvuku na zařízení pro jeho lepší lokalizaci v případě, že se legitimní uživatel nachází v jeho blízkosti. Další možností je volba režimu ztráty, kdy uživatel zadá svoje telefonní číslo, které se prostřednictvím zprávy zobrazí na úvodní obrazovce a případný nálezce jej může na tomto čísle kontaktovat. Tato funkce je přitom aktivní i v případě, kdy je zařízení vypnuto. Potom je po opětovném zapnutí automaticky odeslána e-mailová zpráva o lokalizačních údajích včetně časového razítka poslední lokalizace. Poslední možností vzdáleného přístupu je možnost smazání veškerého obsahu zařízení. Tuto volbu by měl uživatel volit až jako poslední východisko, jelikož při vymazání veškerého obsahu již nelze zařízení dále sledovat ani jej lokalizovat.

V případě tohoto systému jsou navíc při každé lokalizaci následně automaticky zaslány údaje o provedené lokalizaci na předem nastavený e-mail účtu Apple. To je z důvodu, aby zařízení nemohlo být sledováno bez vědomí uživatele. Na následujícím obrázku je v levé části vidět upozornění v případě provedené lokalizace zařízení. V pravé části stejného obrázku je vidět aktivace funkce *Režim ztráty*.

Obrázek 39 Bezpečnostní upozornění systému iOS 7

6.1.6 Bezpečné využívání Bluetooth a Wifi

Pokud se uživatel připojuje k internetu nebo k jiným zařízením prostřednictvím Wifi sítě, měl by se připojovat pouze k důvěryhodným a zabezpečeným sítím. V těchto případech bývá komunikace šifrována pomocí standardních bezpečnostních protokolů. V ostatních

případech se uživatel vystavuje možnosti odposlechu, aniž by si toho musel být vědom. V případě sdílení dat přes rozhraní bluetooth by měl tuto možnost aktivovat pouze v čas přenosu a v případě, kdy toto rozhraní nevyužívá, měl by možnost deaktivovat. Pro komunikaci prostřednictvím webového prohlížeče nebo prostřednictvím poštovního klienta je doporučeno využívat protokolu SSL, kdy jsou data přenášena šifrovaně, a tudíž bezpečně.

6.1.7 Aktualizace

Udržováním aktualizované stavu operačního systému a aplikací se značně snižuje riziko napadení systému. Z hlediska aktualizací jsou důležité zejména aktualizace samotného operačního systému. Aktualizace jsou většinou prováděny pomocí bezdrátové technologie OTA, a to prostřednictvím Wifi nebo mobilní datové sítě. V některých případech jsou prováděny automaticky, kdy je uživatel pouze o provedené aktualizaci informován. V jiných případech je na případnou aktualizaci systémem upozorněn a záleží na něm, jestli aktualizaci provede či nikoli. V případě samotného systému by si měl uživatel sám v nastavení zařízení ověřit, jakou verzi systému jeho zařízení obsahuje a zda je systém aktualizován na nejnovější verzi. V tomto případě máme na mysli verzi, kterou výrobce pro dané zařízení ještě podporuje. Bývá pravidlem, že čím vyšší verze operačního systému je v zařízení nainstalována, tím je zaručena jeho větší bezpečnost.

6.2 Aplikace třetích stran

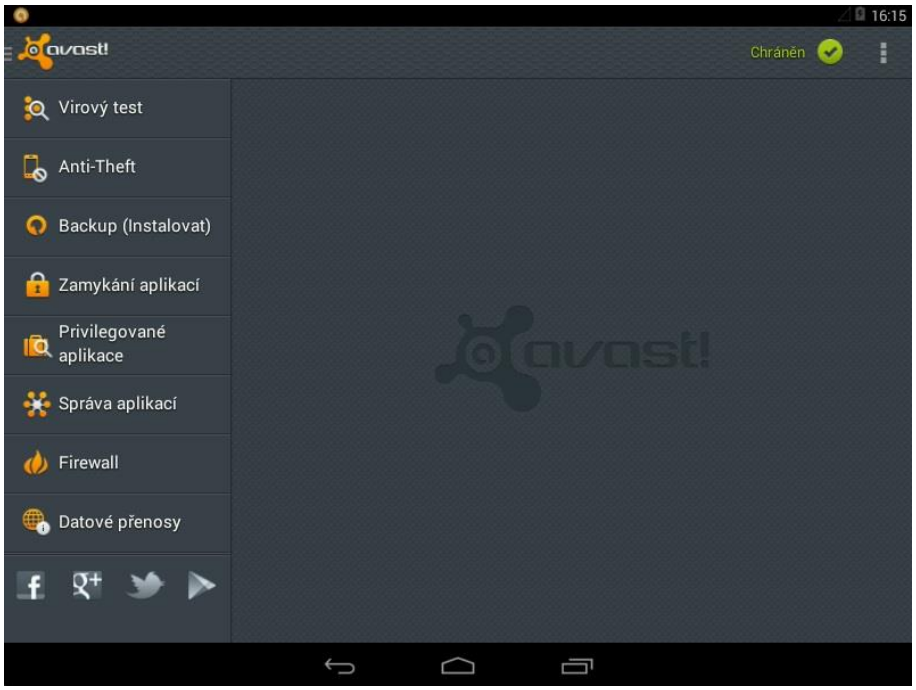
V následující kapitole se zaměříme na vybrané aplikace, které se dají do systému doinstalovat a které tak mohou zvýšit jeho bezpečnost. Mezi tyto aplikace patří zejména antivirové programy a firewally. Ve většině případech se však jedná o aplikace pro systém Android, kdy může být rovněž vyžadováno tzv. root[48]zařízení.

[48] Root zařízení je proces, ve kterém je umožněno uživatelům těchto zařízení využívat privilegovaného režimu. V tomto stavu je na zařízení možné měnit nebo nahrazovat systémové aplikace nebo spouštět funkce, které vyžadují oprávnění správce.

6.2.1 Antiviry

Antivirové programy patří mezi aktivní bezpečnostní prvek, který může značně zvýšit bezpečnost zařízení. Tyto programy jsou dostupné především pro systém Android, který z hlediska své otevřenosti umožňuje nahrát do zařízení jakoukoliv aplikaci z různých neověřených zdrojů. Může se tedy jednat i o aplikaci škodlivou. Taková aplikace, pokud je v zařízení uložena, není systémem pomocí funkce Verify apps detekována. Tato detekce proběhne až v případě pokusu o instalaci. V této kapitole si vyzkoušíme antivirový produkt Avast, který je dostupný pro operační systém Android a je v základní verzi zdarma. Otestujeme, jestli tento antivir dokáže v zařízení škodlivou aplikaci vyhledat ještě před tím, než se uživatel pokusí o její instalaci. K tomuto účelu využijeme opět virtuálního prostředí programu VirtualBox.

Nejprve nainstalujeme z oficiálního obchodu Google Play do zařízení aplikaci *Antivir a ochrana mobilu* z produkce společnosti Avast. Po úspěšné instalaci bude aplikace běžet na pozadí systému, aniž by se musela vždy po zapnutí spouštět. O tom, že je služba aktivní, nás bude informovat ikona v notifikační liště. Jakmile spustíme ikonu antiviru, zobrazí se úvodní obrazovka s hlavní nabídkou produktu (viz. Obrázek 40).



Obrázek 40 Úvodní obrazovka produktu Avast

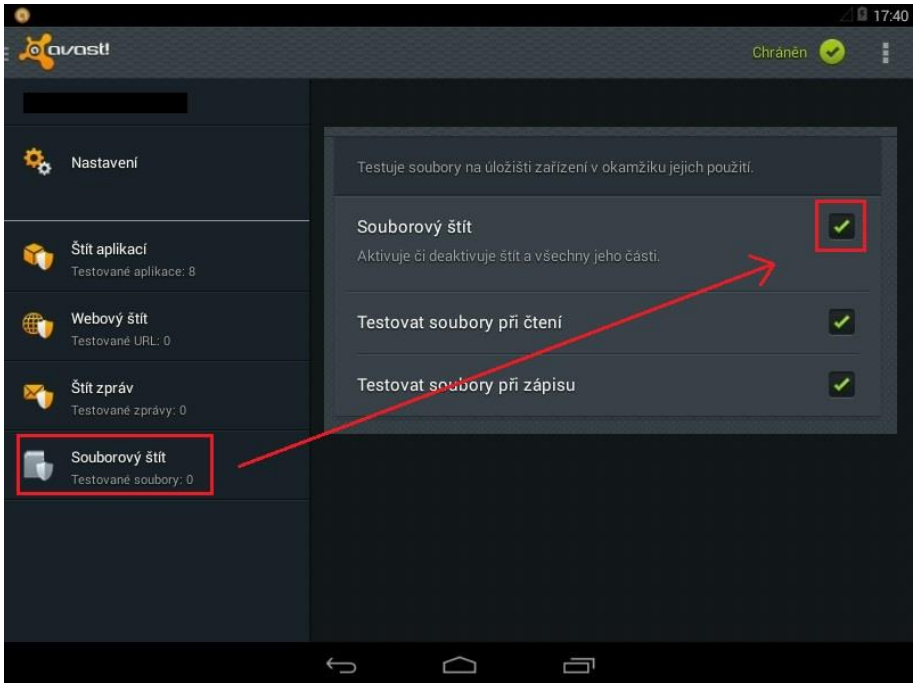
Zde jsou zobrazeny služby, které mohou být z hlediska bezpečnosti velmi přínosné. Některé tyto služby jsou však dostupné pouze jako doplňkové produkty a za registrační poplatek ve verzi Premium.[49]

Mezi tyto produkty patří:

- **Anti-Theft** (chrání zařízení před ztrátou nebo odcizením)
- **Backup** (umožňuje vytvářet zálohy položek, jako jsou kontakty, hovory, SMS zprávy, aplikace atd.)
- **Zamykání aplikací** (umožňuje nastavit, které aplikace budou vyžadovat při svém spuštění zadání uživatelského hesla)
- **Privilegované aplikace** (zobrazuje u jednotlivých aplikací oprávnění, které tyto aplikace používají. Dále řadí tato oprávnění do jednotlivých kategorií, kde zobrazuje informace, jaká aplikace kterou kategorii využívá. Jedná se např. o sledování polohy, čtení identity zařízení, přístup ke kontaktům, přístup ke zprávám, přístup k účtům atd.)
- **Správa aplikací** (zobrazuje veškeré aplikace nainstalované v zařízení a aplikace, které jsou v systému spuštěny)
- **Firewall** (umožňuje nastavovat pravidla pro využívání datové sítě. Pro plnohodnotnou funkci je nutný root zařízení)
- **Datové přenosy** (umožňuje sledování datových přenosů)

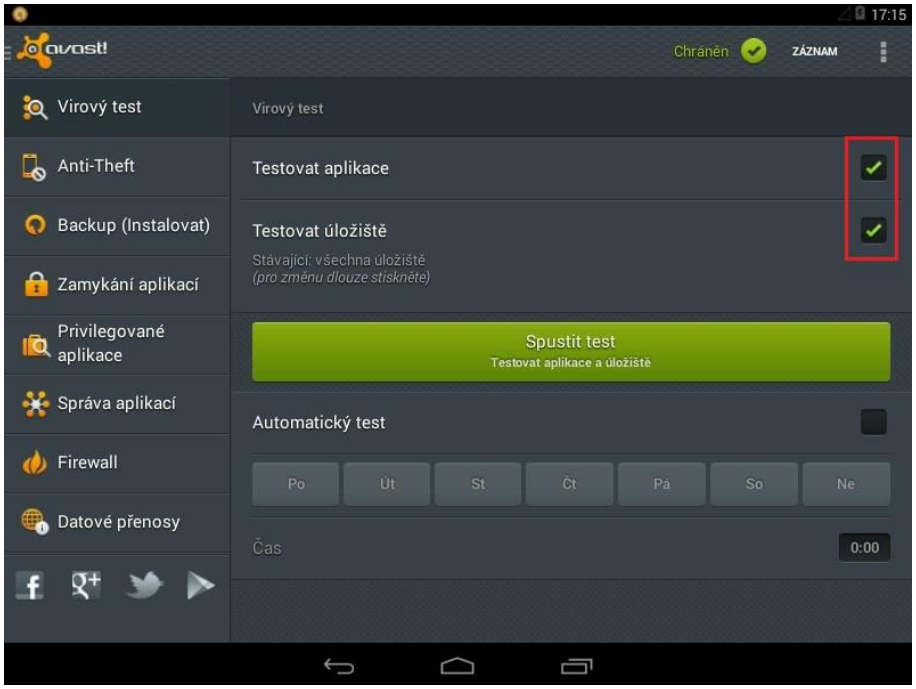
Nyní si vyzkoušíme detekci škodlivé aplikace. Pro tento test využijeme opět naši testovanou aplikaci *Flappy Bird*. Upozorňuji, že v tomto případě byla aplikace do zařízení nahrána pomocí webového prohlížeče z volně přístupného datového úložiště a v této fázi nebyla při základním nastavení antiviru detekována. Proto, aby byla aplikace detekována již ve fázi stáhnutí do zařízení, je nutné aktivovat v nastavení antiviru funkci *Souborový štít* (viz. Obrázek 41).

[49] Prémiové verze některých produktů jsou dostupné po zaplacení stanoveného poplatku a umožňují tak uživateli využívat některé funkce těchto produktů, které nejsou v základní verzi dostupné.



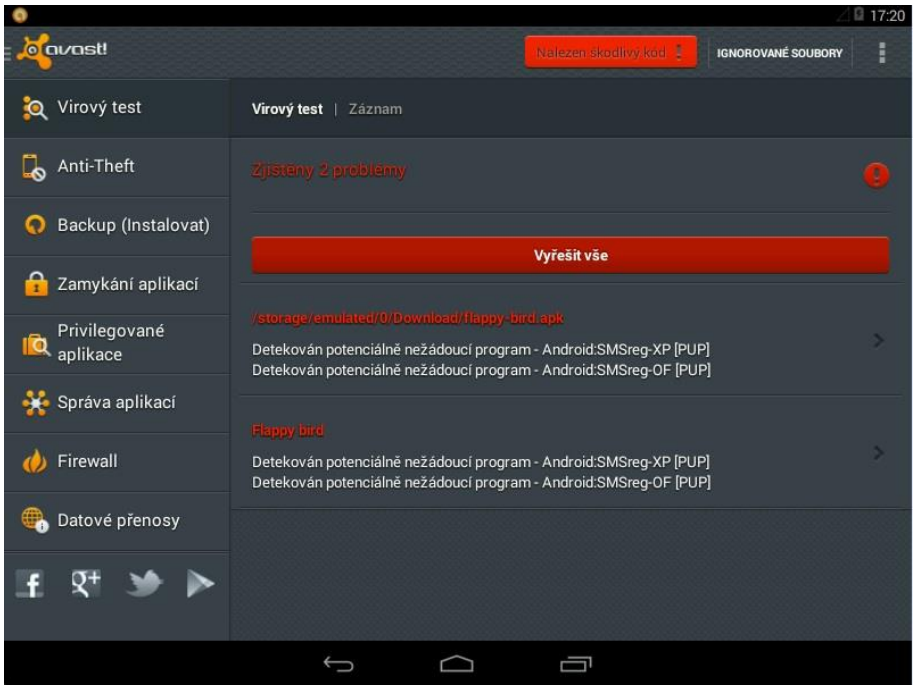
Obrázek 41 Aktivace funkcí souborového štítu

Jakmile máme v zařízení uloženou aplikaci, můžeme přejít k vlastnímu testu detekce. V tomto případě na úvodní obrazovce antiviru přejdeme do složky *Virový test*. Zde zkontrolujeme, zda máme aktivní funkci *Testovat aplikace* a *Testovat úložiště*. Tento krok je zobrazen na následujícím obrázku.



Obrázek 42 Nastavení virového testu

Jakmile máme vše nachystané, klikneme na ikonu *Spustit test* a vyčkáme, až se zobrazí výsledky testu (viz. Obrázek 43).



Obrázek 43 Výsledné hodnoty virového testu

Z výsledku testu je viditelné, že antivir detekoval jak aplikaci, která byla v systému již nainstalovaná, tak aplikaci, která je pouze stažena do úložiště zařízení. V obou případech je možné škodlivou aplikaci ze zařízení odinstalovat nebo z úložiště vymazat. Z toho vyplývá, že uživatel systému Android, který stahuje a instaluje aplikace i z jiných zdrojů, než je služba Google Play, by měl ve svém zařízení mít nainstalován některý z antivirových produktů.

Jak již bylo naznačeno, antivir obsahuje ještě další zajímavé produkty, na které by bylo dobré upozornit. Jedná se o aplikaci *Anti-Theft* a aplikaci *Mobile Backup*. Obě aplikace se musí do zařízení doinstalovat z obchodu Google Play. Aplikace *Anti-Theft* dokáže nastavit zařízení tak, aby jej bylo možné lokalizovat a vypátrat v případě ztráty nebo odcizení. Aplikace *Mobile Backup* vytváří zálohy pro položky kontaktů, hovorů, SMS zpráv, obrázků, audio nebo video souborů a aplikací.

Aby bylo možné využívat služby těchto aplikací, je nutné, aby se uživatel registroval a přihlásil k online účtu Avast. Registrace tohoto účtu je možná zdarma na webových stránkách adresy (https://my.avast.com). Účet je navíc dostupný i pro běžná stolní zařízení obsahující antivirový produkt Avast a je obdobou služby iCloud využívanou pro zařízení

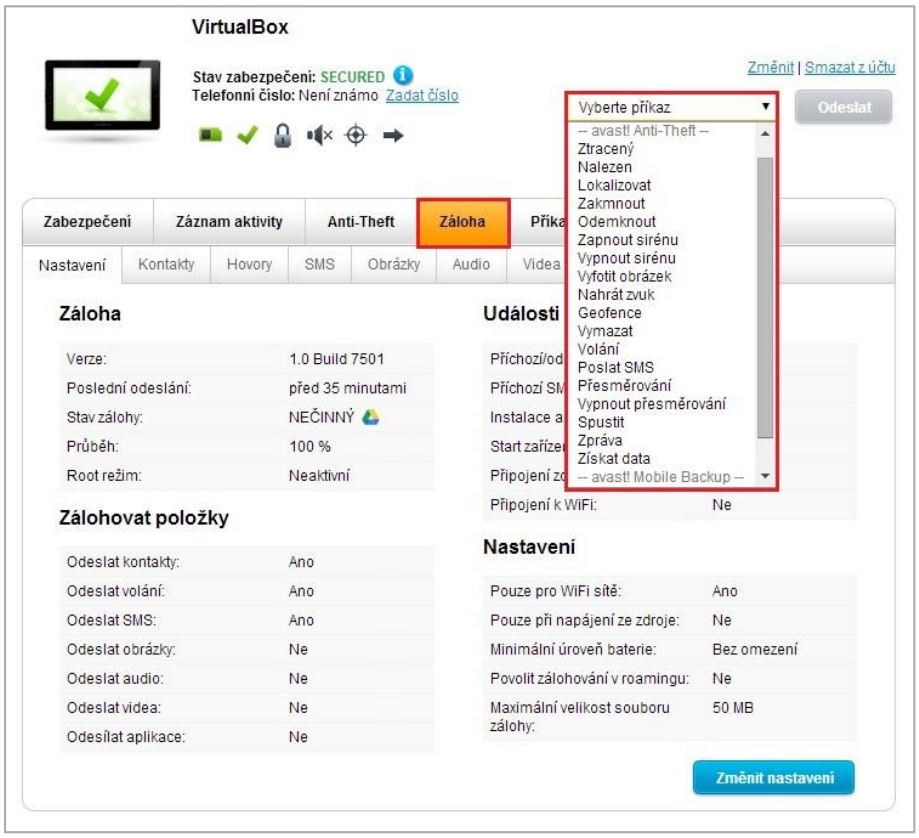
Apple. Po úspěšném přihlášení se zpřístupní dostupné služby a jejich funkce, které je možné využívat. Pro využívání některých funkcí je nutné aktivovat zpoplatněný účet Premium.

Mezi nejvýznamnější funkce patří:

- Lokalizace zařízení na mapovém podkladu
- Zasílání předem nastavených příkazů
- Provést výpis hovorů, zpráv popřípadě kontaktů ze zařízení
- Vytváření záloh
- Aktivace režimu Geofence[50]
- Zamknutí a odemknutí zařízení
- Zaslání zprávy v případě ztráty zařízení
- Vzdálené vymazání zařízení
- Přehrát zvuk na zařízení
- Vyfotit obrázek přední kamerou

Na následujícím obrázku jsou zobrazeny služby účtu Avast s otevřenou záložkou pro vytváření záloh *Backup* a s možností výběru nejrůznějších příkazů služby *Anti-Theft*. Jako detekované zařízení je zde vybrán systém Android, který je poskytován prostřednictvím virtuálního nástroje Virtualbox.

[50] Geofence je režim, který chrání zařízení v rámci předem nastavené lokality (restaurace, bar apod.). V případě, že zařízení opustí vyhrazenou lokalitu, bude tento režim aktivován a zařízení bude označeno jako ukradené.



Obrázek 44 Online služba Avast pro vzdálený přístup

6.2.2 Firewally

Firewally slouží jako kontrolní prvek pro zabezpečení síťového provozu. Pomocí těchto aplikací jsou definována pravidla pro komunikaci mezi zařízením a sítěmi. V případech mobilních zařízení se jedná především o blokaci komunikace jednotlivých aplikací z hlediska přístupu k internetu. Firewally tak mohou být v případě ochrany těchto zařízení značným přínosem. Je nutné si však uvědomit, že takový program, aby fungoval správně, musí mít nad systémem maximální kontrolu. Z tohoto důvodu je pro jejich správnou a plnohodnotnou funkci nutné mít proveden root (OS Android) nebo jailbreak[51] (OS iOS) zařízení, který umožňuje vybraným aplikacím přistupovat k systému s právy administrátora. V tomto upraveném systému lze následně měnit nebo přidávat různá nastavení, která jinak zůstávají pro uživatele skrytá nebo nedostupná. Uživatelům, kteří

[51] Jailbreak je softwarová úprava mobilních zařízení se systémem iOS, která přináší stejné možnosti jako root zařízení pro systém Android. Pomocí jailbreaku je na zařízení vyřazena z provozu ochrana bezpečného spouštěcího procesu a do zařízení tak lze instalovat aplikace i z neoficiálních zdrojů.

nejsou příliš zkušení, se provádět root zařízení nedoporučuje, jelikož takto upravený systém může být podstatně náchylnější v případě útoku.

Na obrázku 45 jsou zobrazena pravidla pro připojení jednotlivých aplikací firewallu zvaném *Root Firewall* určeného pro operační systém Android. V tomto případě se jedná o základní a bezplatnou verzi programu, kdy je umožněno pouze nastavení pravidel připojení pro Wifi nebo 3G síť. Některé firewally však umožňují mnohem větší rozmanitost nastavení. V takových případech je umožněno vytvářet sady pravidel zvlášť pro aplikace, které mají být povoleny nebo které mají být zakázány. Navíc je zde možné nahlédnout do historie výpisu aktivit firewallu nebo do výpisu datových přenosů. Zde je možné vidět, jaké množství dat přenesla určitá aplikace dovnitř nebo ven ze systému, nebo jaká spojení s vnějšími sítěmi tato aplikace využívá.

Obrázek 45 Nastavení pravidel firewallu OS Android

6.3 Bezpečnost firemních zařízení

Ve firemním prostředí jsou kladeny mnohem větší požadavky na bezpečnost, než je tomu v individuálních případech. Tyto požadavky jsou spojeny nejen s fyzickou ochranou, ale také s ochranou duševního vlastnictví každé organizace. Při řešení otázek bezpečnosti bychom měly vycházet z celkové bezpečnostní politiky organizace. Při jejím vytváření je nezbytné, abychom zvážili také hrozby, které plynou z problematiky používání mobilních zařízení. V těchto případech se nezaměřujeme pouze na ochranu samotných zařízení, ale

také na uživatele, kteří tato zařízení využívají. V rámci prosazování bezpečnostní strategie zahrnujeme nejen pravidelné kontroly dodržování bezpečnostních opatření, ale i pravidelná školení uživatelů. Zde je kladen důraz především na to, aby si právě tito uživatelé uvědomovali možná rizika v případě ztráty nebo odcizení zařízení a dodržovali tak základní bezpečnostní opatření.

Mezi možná rizika v případě používání mobilních zařízení v rámci organizace patří:

- **Nedostatečná fyzická kontrola zařízení** (V tomto případě je důležité si uvědomit, že fyzickou kontrolu nad zařízením má především uživatel a nikoli správce dohlížející na informační systémy organizace. Proto je nutné, aby uživatel využíval základní bezpečnostní prvky ochrany zařízení, jako je hlavní heslo nebo šifrování obsahu).
- **Používání nedůvěryhodných zařízení** (Používání důvěryhodných zařízení platí o to víc v případech, kdy ztráta nebo odcizení zařízení představuje značné riziko úniku informací. V tomto případě se jedná o využívání výhradně firemních zařízení nebo o prosazování bezpečnostní strategie i v případech, kdy zařízení jsou ve vlastnictví uživatelů).
- **Využívání nedůvěryhodných sítí** (Zde je nutné zvážit, zda není bezpečnější zcela znemožnit přístup k síťovému připojení nebo alespoň určit zabezpečené sítě, ke kterým se může zařízení připojit. V opačném případě je zařízení vystavěno možným hrozbám jako např. odposlouchávání nebo sledování polohy).
- **Využívání nedůvěryhodných aplikací** (Zde hrozí nebezpečí v případě, kdy lze do zařízení instalovat aplikace z neznámých zdrojů. Tento problém se týká především systému Android, který tuto možnost poskytuje a nelze ji nijak omezit. U systémů iOS a Windows Phone lze v rámci bezpečnostní strategie tuto možnost prostřednictvím programů pro správu systému zakázat).
- **Synchronizace s nedůvěryhodnými systémy** (Každé mobilní zařízení umožňuje synchronizovat data s jinými zařízeními nebo systémy, jako jsou např. cloudové služby. Těmito daty se mohou stát důvěrné informace, jako jsou kontakty, kalendáře, hudba, videa, aplikace nebo e-maily. Zde je nutné si uvědomit, že tímto způsobem mohou uživatelé do zařízení synchronizovat data i z nedůvěryhodných zdrojů. Proto je třeba zvážit, zda není vhodné i tuto možnost zakázat).
- **Používání systému GPS** (Novější zařízení umožňují využívat služeb pro určování polohy. Navíc se stále více aplikací spoléhá na poskytování dat prostřednictvím

GPS souřadnic. Tyto informace přitom mohou být použity pro zahájení cílených útoků, které sdružují uživatele na základě jejich lokalizačních údajů. Těmto hrozbám lze předejít vypnutím systému GPS).

- **Nedostatečné instalace a kontroly aktualizací** (Kontrola a instalace aktualizací je v případě bezpečnosti operačních systémů nezbytná. Připomeňme si, že právě nové verze systému vylepšují nejen jeho vzhled, ale především chyby týkající se jeho bezpečnosti. Kontrolu, zda je systém aktualizován, lze provádět pomocí vybraných aplikací určených pro monitorování systému nebo prostřednictvím vzdálené správy zařízení).

V případech ochrany mobilních zařízení je nutné dodržovat nejen opatření vyplývající z výše uvedených rizik, ale také dodržovat obecné zásady bezpečnosti, které byly popsány na začátku této kapitoly.

Společnosti Google, Apple a Microsoft mají také svoji bezpečnostní politiku, která určuje, zda bude daný operační systém více či méně otevřený či nikoli. Z tohoto hlediska nedoporučuji používat v podnikovém prostředí zařízení se systémem Android, a to z důvodu jeho naprosté otevřenosti a tím pádem i větší zranitelnosti. Naopak v případech systému Windows Phone 8 a iOS 7 jsou tato zařízení vhodná i pro nasazení v rámci organizace. K tomuto účelu společnost Apple i Microsoft vytvořila programy, které mají v rámci podnikového prostředí pomáhat při prosazování a dodržování bezpečnostních zásad a opatření.

V poslední části kapitoly si vyzkoušíme na systému iOS 7 prostřednictvím programu *iPhone Configuration Utility*, jak lze zařízení s tímto systémem bezpečně používat a chránit v rámci organizace. K tomuto účelu použijeme mobilní zařízení iPhone 4 (viz. Obrázek 46), na kterém si ukážeme některá opatření a funkce, které mohou být z hlediska bezpečnosti velmi přínosné. Zde je kladen důraz především na IT správce jednotlivých organizací, kteří by měli v rámci bezpečnostní politiky dbát na dodržování a prosazování bezpečnostní strategie plynoucí z využívání mobilních zařízení.

Obrázek 46 Mobilní zařízení iPhone 4

6.3.1 iPhone Configuration Utility

iPhone Configuration Utility (iPCU) je program vytvořený společností Apple, který je určený pro konfiguraci zařízení se systémem iOS v rámci podnikových systémů. Pomocí tohoto programu lze vytvářet konfigurační profily, které definují, jak bude zařízení v rámci systému pracovat. Po vytvoření konfiguračního profilu je nutné do zařízení tento profil nahrát a následně nainstalovat. To lze učinit těmito možnými způsoby:

- prostřednictvím připojení USB
- prostřednictvím e-mailové přílohy (soubor XML)
- prostřednictvím webového prohlížeče Safari
- bezdrátově prostřednictvím předem nastaveného konfiguračního serveru

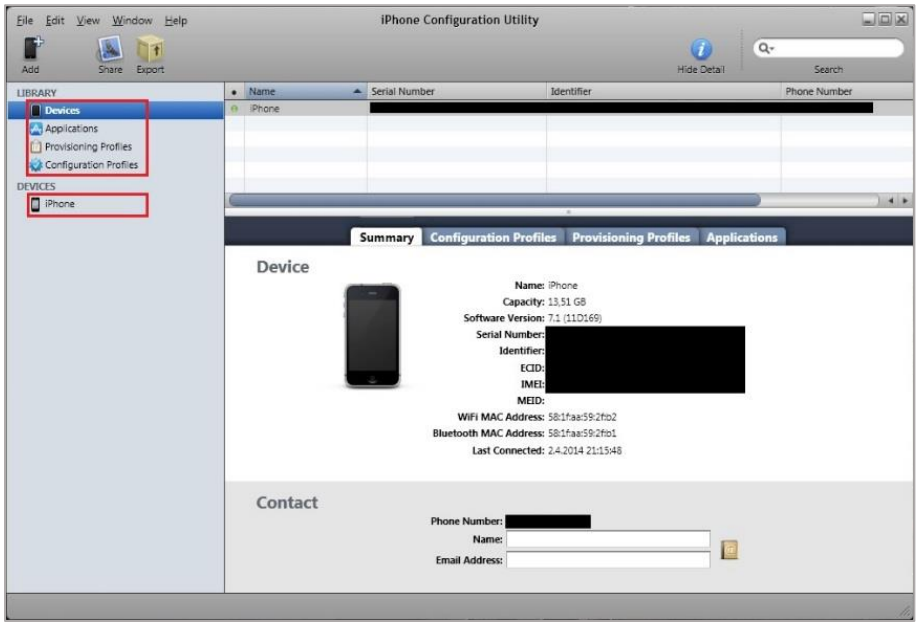
Ve všech těchto případech jsou konfigurační profily kryptograficky podepsané, což znemožní jakoukoliv neoprávněnou změnu v jejich nastavení a omezí jejich použití na konkrétní zařízení.

Pro instalaci programu iPCU je nutné použít systém Windows společnosti Microsoft nebo systém Mac OS X společnosti Apple. V našem případě použijeme systém Windows 7 a nainstalujeme instalační soubor dostupný na adrese (http://support.apple.com/kb/DL1466).

Po spuštění programu se zobrazí úvodní obrazovka, kde se v levém postranním panelu zobrazí knihovna *Library*, která obsahuje tyto čtyři kategorie:

- **Devices** (Zařízení) – záložka obsahuje seznam všech zařízení, která jsou připojena k počítači.
- **Applications** (Aplikace) – v této záložce jsou obsaženy aplikace, které mohou být instalovány v připojených zařízeních. V případech, kdy jsou tyto aplikace vytvořeny v rámci vlastní organizace, může být vyžádán některý z poskytovaných profilů uvedených níže.
- **Provisioning Profiles** (Poskytované profily) – záložka obsahuje seznam profilů schválených v rámci projektu (Apple Developer Connection), kdy je umožněno v zařízení např. provozovat podnikové aplikace, jež nejsou distribuovány prostřednictvím oficiálního obchodu App Store.
- **Configuration Profiles** (Konfigurační profily) – tato záložka obsahuje seznam námi vytvořených konfiguračních profilů, které obsahují předem nastavená pravidla pro vybraná zařízení.

V levém postranním panelu jsou dále zobrazeny informace o aktuálně připojeném zařízení *Devices* prostřednictvím kabelu USB. Zde lze do zařízení instalovat konfigurační profily nebo aplikace přímo z počítače. Zařízení, které je zde zobrazeno, se navíc automaticky přidá do knihovny, což umožní vytvářet profily bez nutnosti fyzického připojení zařízení. Úvodní obrazovka programu iPCU je pro větší přehlednost zobrazena na následujícím obrázku, kde jsou v levé části zobrazeny výše popsané knihovny, pod kterými jsou zpřístupněna aktuálně připojená zařízení.



Obrázek 47 Úvodní obrazovka programu iPCU

1. Vytvoření konfiguračního profilu

Pro vytvoření konfiguračního profilu zvolíme záložku *Configuration Profiles*, kde kliknutím na ikonu *New* přidáme nový konfigurační profil. Po otevření nově přidaného profilu se nám v hlavním okně programu zobrazí přehled dostupných bezpečnostních funkcí a opatření. V další části kapitoly si na vybraných funkcích ukážeme, jak lze konfigurační profil vytvořit a následně zprovoznit v zařízení.

Mezi testovaná bezpečnostní opatření patří:

- General (Obecné nastavení)
- Passcode (Nastavení přístupového hesla)
- Restrictions (Nastavení omezení)
- Wifi (Nastavení Wifi sítí)

General (Obecné nastavení)

V záložce obecného nastavení je umožněno zadat název a identifikátor profilu. Tyto údaje jsou důležité především v případě, kdy je prováděna správa profilů pomocí bezdrátového připojení. Dále je v této nabídce důležitá zejména možnost zabezpečení, kdy je uživateli zařízení znemožněno nainstalovaný profil ze zařízení odebrat, nebo kde je umožněno správci systému nastavit časový interval, kdy bude profil ze zařízení odebrán automaticky.

Passcode (Nastavení přístupového hesla)

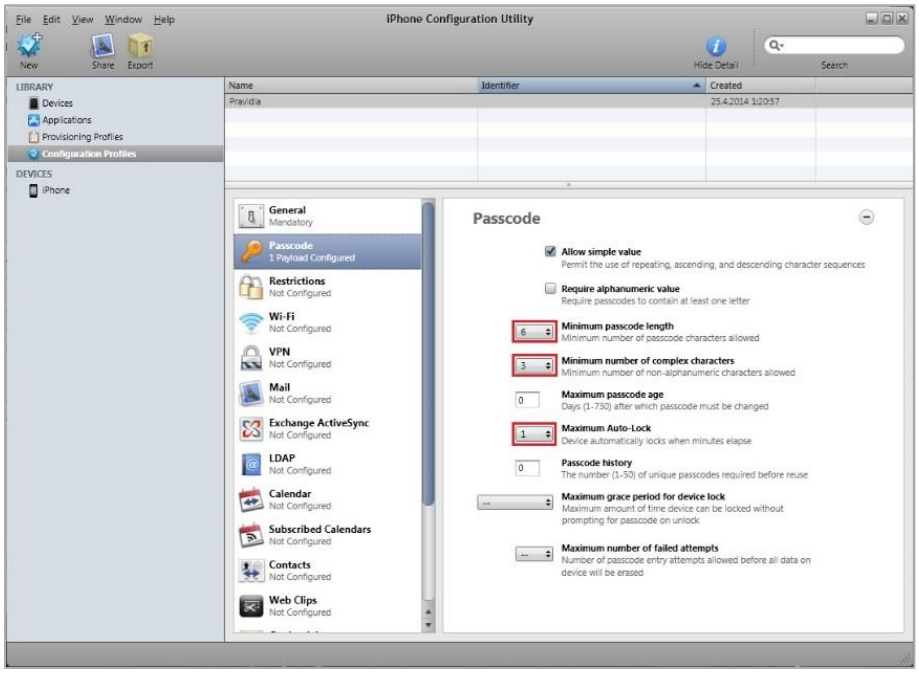
V záložce nastavení přístupového hesla je možné nařídit, zda je zapotřebí zadávat přístupové heslo při každém použití zařízení. Lze zde také určit zásady, které musí být splněny při vytváření nového hesla.

Mezi tyto zásady patří:

- nutnost použití alfanumerických znaků,
- minimální délka hesla,
- minimální počet komplexních znaků,
- znemožnění použití opakujících se hesel,
- časový interval uzamčení přístroje v době jeho nečinnosti,
- možnost vymazání obsahu zařízení po neúspěšných pokusech o zadání hesla (maximálně 10 pokusů).

Záložka pro nastavení přístupového hesla je pro větší přehlednost zobrazena na následujícím obrázku, kde je nastavena volba pro minimální délku hesla, minimální počet

použitých znaků a časový interval doby, kdy se zařízení při nečinnosti automaticky uzamkne.



Obrázek 48 Záložka nastavení přístupového hesla

Restrictions (Nastavení omezení)

V této záložce je správci umožněno nastavit celou řadu omezení, která jsou rozdělena do pěti kategorií:

- Funkce zařízení (v této záložce lze nastavovat omezení jako je např. zákaz instalace aplikací, zákaz používání videohovorů FaceTime, zákaz použití kamery, zákaz používání hlasové asistentky Siri, zákaz nakupovat aplikace atd.)
- Aplikace (umožňuje povolit nebo zakázat použití aplikací YouTube, Safari, iTunes nebo zakázat webovému prohlížeči Safari zobrazit uživateli stránky označené jako podvodné. Záložka umožňuje také blokovat prohlížeči automatické otevírání oken, přijímat soubory cookie nebo zakázat používat JavaScript)
- iCloud (zde je možné povolit nebo zakázat využívání cloudových služeb, jako je synchronizace nebo vytváření záloh)
- Bezpečnost a ochrana osobních údajů (prostřednictvím této záložky lze vypnout zasílání diagnostických údajů do společnosti Apple a zakázat uživateli přijímat nedůvěryhodné certifikáty)
- Hodnocení obsahu (nastavuje pravidla pro hodnocení filmů, televizních pořadů a aplikací)

Na následujícím obrázku je zobrazeno, jak lze zablokovat možnost instalace aplikací z obchodu App Store, možnost používání kamery a možnost používání aplikace YouTube.

Obrázek 49 Záložka nastavení omezení

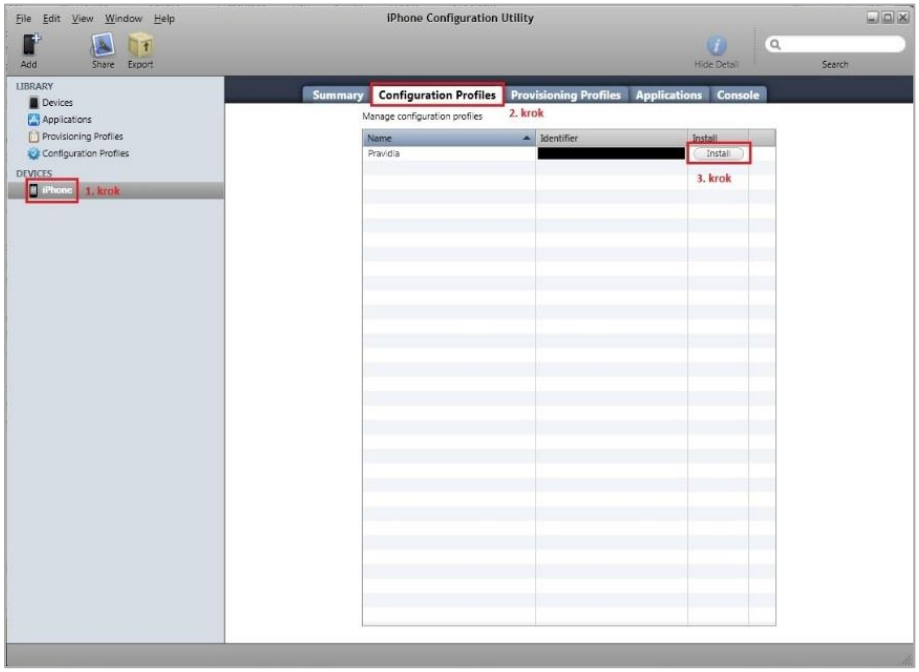
Wifi (Nastavení Wifi sítí)

V záložce Wifi lze spravovat nastavení pro připojení zařízení k bezdrátovým podnikovým sítím, ke kterým se může zařízení připojit. Jsou zde zpřístupněny volby, jako je přístupové heslo nebo způsob šifrování přenosu (viz. Obrázek 50).

Obrázek 50 Záložka nastavení sítě Wifi

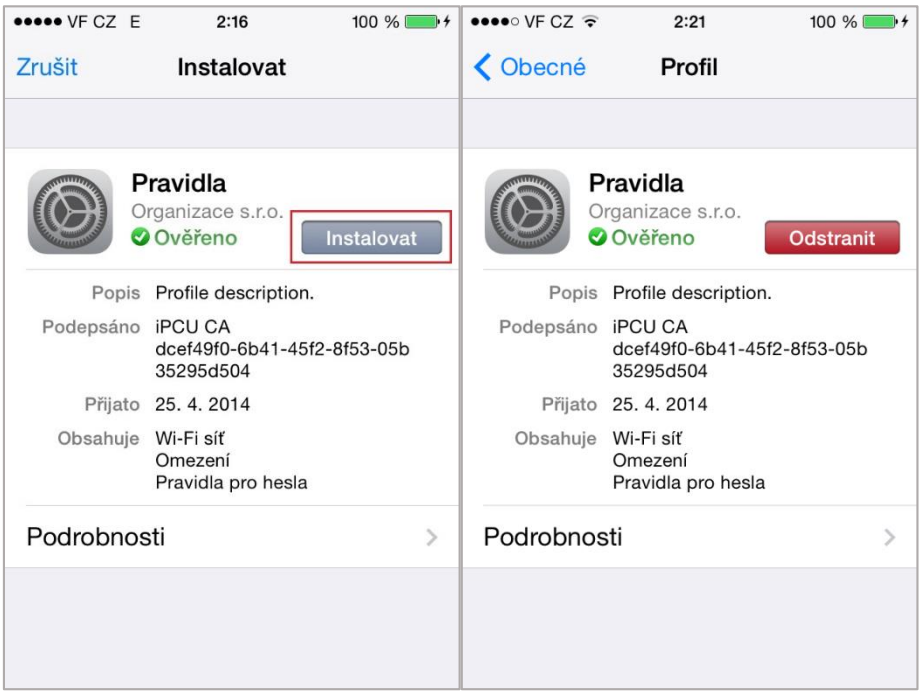
2. Instalace konfiguračního profilu

Jakmile máme konfigurační profil nastavený a připravený k instalaci, můžeme instalační proces spustit a profil tak nainstalovat do zařízení. V tomto případě přepneme do složky *Devices*, kde vybereme určené zařízení a zvolíme záložku *Configuration Profiles*. Odtud už stačí pouze spustit instalaci konfiguračního profilu do zařízení. Nezapomeňme, že je nutné mít v položce *Identifier* zvolené identifikační číslo cílového zařízení. Jednotlivé kroky pro instalaci jsou zobrazeny na obrázku 51.



Obrázek 51 Instalace konfiguračního profilu v zařízení

Po kliknutí na položku *Install* se spustí proces instalace, kdy je nutné tento proces v zařízení potvrdit tlačítkem *Instalovat*. Tento krok je zobrazen na obrázku 52 vlevo. V pravé části stejného obrázku je vidět volba *Odstranit*, pomocí které je možné ze zařízení konfigurační profil uživatelem odebrat a zařízení tak provozovat jako by profil nebyl vůbec instalován. Tato volba je však dostupná pouze v případě, že jsme v záložce obecného nastavení konfiguračního profilu nechali aktivní položku umožňující odstranění profilu.



Obrázek 52 Proces instalace konfiguračního profilu

Případnou kontrolu jednotlivých profilů v zařízení lze provést tak, že po kliknutí na ikonu *Nastavení* v záložce *Obecné* zvolíme položku *Profil*. Zde jsou zobrazeny všechny dostupné konfigurační profily spolu s výpisem bezpečnostních pravidel a opatření, která obsahují.

Závěrem této kapitoly lze konstatovat, že zařízení se systémem iOS jsou z hlediska nasazení v podnikovém prostředí zcela vyhovující. Tato bezpečnostní strategie by mohla být použita i v případě domácího prostředí, kdy lze dětem omezit nebo zcela zakázat využívání některých funkcí systému. V tomto případě se může jednat např. o funkci nákupu nových aplikací, prohlížení webového obsahu nebo instalaci aplikací z obchodu App Store.

ZÁVĚR

Bezpečnost dat, ať už v klasických nebo mobilních zařízeních, je poslední dobou velmi důležitou a diskutovanou tematikou. Této skutečnosti napomáhá především velký rozmach v oblasti mobilních technologií, kdy je na trhu stále více zařízení, které svými funkcemi dokáží zcela nahradit klasické počítače. Například prostřednictvím dnešních mobilních telefonů se můžeme připojit k veřejné síti téměř z kteréhokoliv místa nebo z takového zařízení můžeme vyřizovat veškeré finanční transakce. Dalším faktorem je, že čím dál více uživatelů svěřuje svým zařízením stále více důvěrných informací, ať už ve formě kontaktů, fotografií, přihlašovacích nebo jiných údajů. Není se tedy čemu divit, když jsou na tato zařízení kladeny stále větší požadavky z hlediska bezpečnosti. V tomto případě nemáme na mysli pouze fyzickou bezpečnost zařízení, ale především bezpečnost, která se týká ochrany osobních údajů. Ta je dána především architekturou operačních systémů, které tato data chrání před neoprávněnou manipulací.

V diplomové práci bylo úkolem seznámit čtenáře s problematikou malwaru na mobilních zařízeních. V teoretické části jsem se zaměřil na vysvětlení základní charakteristiky malwaru, kde byly představeny jeho nejběžnější kategorie a také jeho neznámější představitelé z období historie. Dále zde byly podrobněji popsány operačních systémy Android, iOS a Windows Phone 8. U těchto systémů byla rozebrána jejich historie, architektura a také zde byly podrobněji popsány bezpečnostní prvky, které tyto systémy chrání. Právě tyto bezpečnostní prvky se u těchto systémů neustále vyvíjí a zdokonalují, což lze vidět s příchodem každé nové verze operačního systému.

V praktické části práce jsem se podrobněji zaměřil především na operační systém Android, kde jsem za pomocí virtuálního prostředí VirtualBox a škodlivé aplikace otestoval jeho novou funkci, která má za úkol tento systém chránit před škodlivým malwarem. V tomto případě se jedná o funkci Verify apps a jak bylo ukázáno, je tato kontrola dostatečná pouze v případě, kdy se uživatel chystá aplikaci nainstalovat ve svém zařízení. V ostatních případech, kdy je aplikace pouze stahována do paměti nebo v ní uložena, je tato ochrana neúčinná. Dále zde byla provedena analýza škodlivé aplikace prostřednictvím vybraných webových služeb, kde jsem se zaměřil na některá oprávnění, kterých tato aplikace využívá. V tomto případě bylo zjištěno, že u systému Android je právě kontrola jednotlivých oprávnění zobrazujících se při instalaci aplikace velmi důležitá, a to už jen proto, že kontrolu provádí samotný uživatel a může vést k prvním náznakům, že je aplikace

škodlivá. Tuto skutečnost si však ne každý uživatel uvědomuje a ve většině případech tato oprávnění bezmyšlenkovitě odsouhlasí. V poslední části práce jsem se zaměřil na další možnosti zabezpečení, kde jsem využil některých produktů třetích stran, které mohou být značným přínosem především pro uživatele ne tolik bezpečného systému Android. V tomto případě jsem otestoval antivirový produkt Avast, kde jsem pomocí obrázků ukázal, jak mohou být jeho některé funkce prospěšné. To platí převážně pro uživatele, kteří do svých zařízení stahují aplikace z neoficiálních zdrojů. Z testu tak bylo vidět, že antivir dokáže vyhledat škodlivou aplikaci i v případě, že je pouze stažena v úložišti zařízení a tuto aplikaci dokáže nejen identifikovat, ale také odstranit. V poslední části jsem poukázal na možnosti zabezpečení firemních zařízení. Zde jsem za použití speciálního softwaru a systému iOS předvedl, jak lze chránit zařízení v podnikovém prostředí.

Závěrem lze konstatovat, že záleží především na každém uživateli, jakému operačnímu systému dá při koupi nového zařízení přednost. Zda zvolí spíše otevřený, a tudíž méně bezpečný systém Android, nebo zvolí spíše systémy iOS nebo Windows Phone 8, které jsou sice systémy uzavřenými, ale z hlediska své bezpečnosti jsou vhodné i pro použití ve firemním prostředí. Každopádně systém Android není v současné době z hlediska bezpečnosti nijak zvlášť chráněný, a proto ho nedoporučuji používat v organizacích, kde by mohlo dojít k úniku důvěryhodných informací. Tato skutečnost platí i v případě, že jeho novější verze poskytují ochranu prostřednictvím nové funkce Verify apps, kdy tato kontrola ještě není zcela spolehlivá a uživatel by se na ni neměl stoprocentně spoléhat. Tato skutečnost je dána především tím, že každá aplikace není bezpečnostním týmem kontrolována ještě před uvedením na trh, ale až v případě, kdy se vyskytnout pochybnosti, že by se mohlo jednat o škodlivou aplikaci. V praxi to znamená značné riziko, jelikož aplikace mohou být z obchodu vyřazeny až se značným zpožděním. To je velká nevýhoda oproti systému iOS nebo Windows Phone, kde každou aplikaci kontroluje tým odborníků ještě před tím, než je aplikace zpřístupněna veřejnosti. V těchto případech má uživatel mnohem větší jistotu, že je aplikace opravdu nezávadná. Další výhodou systémů iOS a Windows Phone 8 je skutečnost, že zpravidla neumožňují aplikace do svých zařízení instalovat z jiných zdrojů, než jsou oficiální obchody. Na této ochraně se výrazným způsobem podílejí dva důležité bezpečnostní prvky, jako je bezpečnostní spouštěcí proces a systémová personalizace. Tyto bezpečnostní prvky však vyžadují speciální hardwarové zabezpečení v každém zařízení.

ZÁVĚR V ANGLIČTINĚ

Data immunity, considering both, classical or mobile is a quite often discussed issue nowadays. The main reason of this is a fast growing demand in range of mobile technology, when more and more equipments which are able to fully replace classical IT technology appear in the market. For example, it is common to connect to a public network almost everywhere through a cell phone as well as financial transactions are able to be solved with such devise. Another factor is a fact, that more users provide personal information to these devices by saving photographs, contacts, access and others data. It is no wander then, that more security demands are expected from these equipments. In this case, it is not considered just its physical security, but mainly security dedicated to personal data. This one is set mainly with architecture of operation system, which secures data from an unauthorised manipulation.

The purpose of the work has been about to introduce a reader with a malware problematic in mobile devices. I have made a focus on a basic malware´s characteristic explanation in the theoretical part, where its the most common categories as well as the most know representatives in the history were introduced. Also the further description of Android, iOS and Windows Phone 8 operation systems has been made. History and architecture of these systems has been analysed here as well as security elements which secure these system were described. These are the elements, which are continuously involving and improving, which is being shown with income of every new operation system´s versions.

I have made a further focus mainly on Android operational system, where I have tested its new function, which is supposed to secure the system from a harmful malware. The test has been made with help of the VirtualBox virtual environment and a harmful application. In this case, it was about Verify apps´s function and it was proven that this inspection is complement just in a time being when an application is being installed into a device. In other cases, when an application is being only downloaded into a memory or saved, the security is not being valid. Then, an analyse harmful application through chosen web services has been made, where I have made a focus on some of authorities, which the application is using. In this case, it has been found, that in Android systems, a control of single licence which is being shown meanwhile an application installation is very important. That is because a control is being made just by a user and it can lead to first

sings, that an application is harmful. No every user is aware of such danger, therefore he would allow an access with no considering it.

The last part employs itself with another way of security, where I have used some product of the third part, which can have a large benefit mainly for an owner of not really safe Android system. In this case, I have tested an antivirus product Avast, where I have used the animation to displayed how much can some of its function be useful. This mainly holds for users, who use unofficial sources for downloading. The test has showed, that antivirus is able to find a harmful application also in a case, when the application has been downloaded only in a heap of a device. It is not just able to identify the application, but also remove it. I have mentioned possibilities of commercial equipment securing in the last part of the work and I have also presented the way how it is possible to secure devices in commercial environment by use of special software and iOS system.

On the end, it can be claimed, that it is about every user, which operational system he prefers. If he will decide for an open system, therefore less secure Android or will chose iOS or Windows Phone 8, which are closed systems, but are suitable for commercial environment for their security. Anyway, the Android system in not having a strong security nowadays, therefore it is not recommended to be used in organizations, where an outflow of secret information would appear. This matter of fact also applies in cases where new versions provide a security through of a new Verify apps function. This control is not being very faithful yet, therefore a user should not trust this function for hundred percent. This reality has been set by the fact, that not every application is being controlled by a security team before its release on market, but after an impeachment of its credibility. It means a big risk for practise, because these dangerous applications are then being removed with en extensive delay. This is a big demerit in comparison with iOS or Windows Phone 8, where each application is being controlled by a team of professionals, before its access to public. In these cases, the user is having much bigger confidence, that an application is not harmful. Another advantage of iOS and Windows Phone 8 is the fact that it is not allowed to install applications into devices from different sources of the official shops. Two important security elements are participating with this security. Protective starting process and system´s personalisation. These security elements require special hardware security in every device.

SEZNAM POUŽITÉ LITERATURY

[1] DUNHAM, Ken. *Mobile malware attacks and defense*. Burlington, MA: Elsevier, c2009, xxv, 409 p. ISBN 15-974-9298-1.

[2] HIMANSHU DWIVEDI, Chris Clark. *Mobile application security*. New York: McGraw-Hill, 2010. ISBN 978-007-1633-574.

[3] F-SECURE CORP. Mobile Threat Report Q4 2012. In: *Mobile Threat Report Q4 2012* [online]. 2013, 07.03.2013 [cit. 2014-02-24]. Dostupné z: <http://www.f-secure.com/static/doc/labs_global/Research/Mobile%20Threat%20Report%20Q4%202012.pdf>.

[4] STRATEGY ANALYTICS. Android Captured 79% Share of Global Smartphone Shipments in 2013. In: [online]. [cit. 2014-02-27]. Dostupné z: http://blogs.strategyanalytics.com/WSS/post/2014/01/29/Android-Captured-79-Share-of-Global-Smartphone-Shipments-in-2013.aspx.

[5] HOOG, Andrew. *Android forensics: investigation, analysis, and mobile security for Google Android*. Amsterdam: Elsevier, c2011, xix, 372 s. ISBN 978-1-59749-651-3.

[6] SMITH, Dave a Jeff FRIESEN. *Android recipes: a problem-solution approach*. New York: Distributed to the book trade worldwide by Springer Science Business Media, c2011, xiii, 442 p. ISBN 978-1-4302-3414-2.

[7] Dashboards: Platform Versions. OPEN HANDSET ALLIANCE. *Android Developer* [online]. 2014 [cit. 2014-03-02]. Dostupné z: <https://developer.android.com/about/dashboards/index.html#Platform>.

[8] AUTHORS, Anmol Misra. *Android security: attacks and defenses*. Boca Raton, Fla: CRC Press, 2013. ISBN 14-398-9646-1.

[9] SIX, Jeff. *Application security for the Android platform*. 1st ed. Sebastopol, CA: O'Reilly, 2011c2012, x, 97 p. ISBN 14-493-1507-0.

[10] Android Security Overview. OPEN HANDSET ALLIANCE. *Android Open Source Project* [online]. 2014 [cit. 2014-03-02]. Dostupné z: <http://source.android.com/devices/tech/security/>.

[11] RAI, Pragati Ogal. *Android application security essentials: write secure Android applications using the most up-to-date techniques and concepts*. Birmingham: Packt Pub, 2013. ISBN 978-184-9515-603.

[12] KAPLAN, Dan. Google using custom malware scanner for Android apps. In: *SC Magazine: IT Security News and Security Product Reviews* [online]. 2012 [cit. 2014-03-09]. Dostupné z: http://www.scmagazine.com/google-using-custom-malware-scanner-for-android-apps/article/226068/.

[13] THE VERGE. *IOS: A visual history* [online]. 2013, 16. 9. 2013 [cit. 2014-03-11]. Dostupné z: http://www.theverge.com/2011/12/13/2612736/ios-history-iphone-ipad.

[14] APPLE INC. *IOS Technology Overview: About the iOS Technologies* [online]. 2013, 18. 9. 2013 [cit. 2014-03-12]. Dostupné z: https://developer.apple.com/library/ios/documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/Introduction/Introduction.html.

[15] HOOG, Andrew. *IPhone and iOS forensics: investigation, analysis, and mobile security for Apple iPhone, iPad, and iOS devices*. Amsterdam: Elsevier, c2011, xv, 310 s. ISBN 978-1-59749-659-9.

[16] APPLE INC. *IOS Security* [online]. 2012 [cit. 2014-03-12]. Dostupné z: http://www.apple.com/ipad/business/docs/iOS_Security_Oct12.pdf.

[17] WHITECHAPEL, Andrew a Sean MCKENNA. *Windows Phone 8 development internals*. Sebastopol, California: O'Reilly Media, Inc., 2013, xxx, 1009 p. ISBN 07-356-7623-2.

[18] MICROSOFT CORPORATION. *Windows Phone 8 Security Guide* [online]. 2013 [cit. 2014-03-25]. Dostupné z: http://www.windowsphone.com/en-us/business/security.

[19] CARO ORGANIZATION. *CARO Naming Scheme: NameSyntax* [online]. 2014 [cit. 2014-04-11]. Dostupné z: http://www.caro.org/naming/namesyntax.html.

[20] SIKORSKI, Michael. *Practical malware analysis: the hands-on guide to dissecting malicious software*. San Francisco: No Starch Press, c2012, xxxi, 766 s. ISBN 978-1-59327-290-6.

[21] ARAS, Kai. STUTTGART MEDIA UNIVERSITY. *Jailbreaking iOS: How an iPhone breaks free* [online]. 2014 [cit. 2014-04-27]. Dostupné z: http://www.slideshare.net/ka010/jailbreaking-ios.

[22] UNIWERSITY OF TWENTE. *XNU: a security evaluation* [online]. 2012 [cit. 2014-04-27]. Dostupné z: http://reverse.put.as/wp-content/uploads/2011/06/XNU_-a-security-evaluation-Daan_Keuper_2012-12-14-xnu.pdf.

[23] MICROSOFT CORPORATION. *Windows Phone: Historie aktualizací Windows Phone 7* [online]. 2014 [cit. 2014-04-28]. Dostupné z: http://www.windowsphone.com/cs-cz/how-to/wp7/basics/update-history.

[24] MICROSOFT CORPORATION. *Windows Phone: Historie aktualizací Windows Phone 8* [online]. 2014 [cit. 2014-04-28]. Dostupné z: http://www.windowsphone.com/cs-cz/how-to/wp8/basics/windows-phone-8-update-history.

[25] KOČÍ, Mirek. *Svět aplikací: Instalujeme vývojářský update Windows Phone 8 GDR 3* [online]. 2013, 17.10 [cit. 2014-04-28]. Dostupné z: http://svetaplikaci.tyden.cz/instalujeme-vyvojarsky-update-windows-phone-8-gdr3/.

[26] TECHSPOT INC. *HP ProBook 4720S - Intel Core i5* [online]. 2014 [cit. 2014-04-28]. Dostupné z: http://www.techspot.com/products/laptops/hp-probook-4720s-intel-core-i5.17687/.

[27] MILLER, Charles. *IOS hacker's handbook*. Indianapolis, IN: Wiley, c2012, xx, 388 p. ISBN 11-182-0412-3.

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

AES	Advanced Encryption Standard
API	Application Programming Interface
APK	Android Application Package
ASLR	Address Space Layout Randomization
BIOS	Basic Input Output System
CLR	Common Language Runtime
CORA	Computer Antivirus Research Organization
CPU	Central Processing Unit
DFU	Device Firmware Update
DMA	Direct Memory Access
DSA	Digital Signature Algorithm
EAS	Exchange Active Sync
ECID	Electronic Chip ID
FAT	File Allocation Table
FUP	Fair Use Policy
GID	Group Identifier
GPS	Global Position System
GPU	Graphic Processing Unit
HCE	Host Card Emulation
HTML	HyperText Markup Language
HTTPS	HyperText Transfer Protocol Secure
IDEP	iOS Developer Enterprise Program
IPC	Inter-Process Comunication
iPCU	iPhone Configuration Utility

IRM	Information Rights Management
IT	Information Technology
J2ME	Java 2 Micro Edition
JVM	Java Virtual Machine
LLB	Low Level Boot loader
MD5	Message Digest 5
MDM	Mobile Device Management
MMS	Multimedia Messaging Service
NFC	Near Field Communication
NT	New Technology
NTFS	New Technology File System
OHA	Open Handset Alliance
OS	Operating System
OTA	Over The Air
P2P	Peer To Peer
PBKDF2	Password Based Key Derivation Function
PIE	Position Independent Executable
PIN	Personal Identification Number
RAM	Random Access Memory
RNG	Random Number Generation
ROM	Read Only Memory
RSA	Rivest Shamir Adleman
SDK	Software Development Kit
SDL	Security Development Lifecycle
SHA	Secure Hash Algorithm
SIM	Subscriber Information Module

SIS	Symbian Installation File
SMS	Short Message Service
SQL	Structured Query Language
SSL	Secure Sockets Layer
TPM	Trusted Platform Module
TSL	Transport Layer Security
UEFI	Unified Extensible Firmware Interface
UID	Unique Identifier
USB	Universal Serial Bus
VDI	VirtualBox Disk Image
VM	Virtual Machine
VPN	Virtual Private Network
XAML	Extensible Application Markup Language
XML	Extensible Markup Language

SEZNAM OBRÁZKŮ