Prostředí pro testování bezpečnosti aplikací pro iOS

Lukáš Talacek

Bakalářská práce
2020

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně

Fakulta aplikované informatiky

Ústav informatiky a umělé inteligence

Akademický rok: 2019/2020

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení:	**Lukáš Talacek**
Osobní číslo:	**A17569**
Studijní program:	**B3902 Inženýrská informatika**
Studijní obor:	**Softwarové inženýrství**
Forma studia:	**Prezenční**
Téma práce:	**Prostředí pro testování bezpečnosti aplikací pro iOS**
Téma práce anglicky:	**An Environment for the Security Testing of iOS Applications**

Zásady pro vypracování

1. Zpracujte teoretický základ na téma testování SW.
2. Zpracujte architekturu iOS s ohledem na její principy a funkcionalitu.
3. Vysvětlete pojem Jailbreak.
4. Vytvořte modelový příklad testovacího prostředí.

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

Seznam doporučené literatury:

1. VÁVRŮ, Jiří. *IPhone: vývoj aplikací*. Praha: Grada, 2012. Průvodce (Grada). ISBN 978-80-247-4457-5.
2. BUREŠ, Miroslav, Miroslav RENDA, Michal DOLEŽEL, Peter SVOBODA, Zdeněk GRÖSSL, Martin KOMÁREK, Ondřej MACEK a Radoslav MLYNÁŘ. *Efektivní testování softwaru: klíčové otázky pro efektivitu testovacího procesu*. Praha: Grada, 2016. Profesionál. ISBN 978-80-247-5594-6.
3. CRESPO, Laura. Best Automated Testing Tools for iOS. In: *Bugfender* [online]. Mobile Jazz, 2018, 29.10.2018 [cit. 2019-11-25]. Dostupné z: https://bugfender.com/blog/best-automated-testing-tools-ios/
4. *OWASP Foundation* [online]. US: OWASP Foundation, 2019 [cit. 2019-11-22]. Dostupné z: https://www.owasp.org/index.php/Main_Page
5. IOS App Testing Tutorial: Manual & Automation. In: *Guru99* [online]. Guru99, 2019, 29.10.2018 [cit. 2019-11-25]. Dostupné z: https://www.guru99.com/getting-started-with-ios-testing.html
6. AHAMAD, Sahil. Basic iOS Apps Security Testing lab. In: *Medium* [online]. A Medium Corporation, 2019, 22.9.2018 [cit. 2019-11-25]. Dostupné z: https://medium.com/@ehsahil/basic-ios-apps-security-testing-lab-1-2bf37c2a7d15

Vedoucí bakalářské práce: **Ing. Lukáš Králík**
Ústav počítačových a komunikačních systémů

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

L.S.

doc. Mgr. Milan Adámek, Ph.D.
děkan

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

Ve Zlíně dne 9. prosince 2019

Prohlašuji, že

- beru na vědomí, že odevzdáním 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 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 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 připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování 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 bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem 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 bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne …….

podpis diplomanta

ABSTRAKT

Tato bakalářská práce má za cíl přiblížit problematiku testování aplikací na platformě iOS. Cílem bylo připravit kompletní prostředí, což zahrnuje speciálně připraven telefon, i počítač, a osvětlit tuto činnost i pro nezasvěcené čtenáře. Tvorba zahrnuje průzkum takových řešení a následně jejich zhotovení. Na závěr práce zhodnotí dosažené výsledky a jejich aktuálnost a využitelnost.

Klíčová slova: iOS, testování softwaru, Jailbreak

ABSTRACT

This bachelor thesis aims to approach the problematics of application testing on the iOS platform. The was to prepare a complete environment, which includes a specially prepared telephone and computer, and to illuminate this activity even for uninitiated readers. The creation includes the research of such solutions and their subsequent construction. At the end of the work assesses the achieved results and their topicality and usability.

Keywords: iOS, software testing, Jailbreak

Zde bych rád poděkoval pár lidem. Děkuji panu Ing. Lukáši Králíkovi, za odborné vedení práce a cenné rady, díky kterým jsem byl schopen tuto práci zdárně zkompletovat. Děkuji své partnerce a rodině, za jejich trpělivost a oporu po celou dobu studia, a hlavně při psaní mé bakalářské práce.

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

OBSAH

ÚVOD **9**

I TEORETICKÁ ČÁST **10**

1 ZÁKLADY TESTOVÁNÍ SOFTWARE **11**

1.1 CO JE TO TESTOVÁNÍ A PROČ SE DĚLÁ 11

1.2 KRÁTKÁ HISTORIE 11

1.3 TESTOVACÍ METODY 12

1.3.1 Statické a dynamické testování 12

1.3.2 Black box a white box 12

1.3.3 Regresní testování 13

2 ARCHITEKTURA IOS **14**

2.1 FUNKCE IOS 14

2.1.1 Vrstvy systému 15

2.2 AKTUÁLNÍ IOS 17

2.2.1 iOS 13 17

2.3 ROZDÍLY SYSTÉMŮ ANDROID A IOS, JEJICH FUNKCIONALITA 18

3 JAILBREAK **20**

3.1 HISTORIE JAILBREAKU 20

3.2 TYPY 21

3.2.1 Tethered 21

3.2.2 Untethered 21

3.2.3 Semi-Tethered 21

3.2.4 Semi-Untethered 21

3.3 AKTUÁLNÍ STAV JAILBREAKU 21

II PRAKTICKÁ ČÁST **22**

4 TESTOVACÍ PROSTŘEDÍ **23**

4.1 XCODE 23

4.1.1 Emulátory 24

4.2 PŘÍPRAVA TELEFONU 25

4.2.1 Jailbreak 25

4.2.2 Cydia tweaky 27
4.3 PŘÍPRAVA POČÍTAČE 29
4.3.1 Instalace programů 29
4.3.1.1 iExplorer 29
4.3.1.2 Cydia Impactor 30
4.3.1.3 Burp Suite 30
4.3.2 Externí nástroje pro iOS 37
4.4 SHRNUTÍ 48
ZÁVĚR 49
SEZNAM POUŽITÉ LITERATURY 50
SEZNAM OBRÁZKŮ 52
SEZNAM TABULEK 53

ÚVOD

V teoretické části práce přiblíží pojem testování software. Je to velmi důležité téma, co se týče vývoje software a v posledních letech a nabírá na rozšířenosti. Profesionálové v tomto oboru dokáží společnostem ušetřit nespočet financí a času, jejichž výdaj na dodatečné úpravy a opravy produktů může být pro spoustu týmů opravdu fatální.

Ke smyslu této práce je také třeba zasvětit nezkušené do světa iOS. Ne však pouze na uživatelské úrovni, jde se více do hloubky. Nebezpečí je totiž v nižších vrstvách systému, nejen na „povrchu“. Porovnání s největším konkurentem na trhu – Androidem – čeká v následujících kapitolách také. Nejedná se však o zhodnocení, který systém je lepší, slouží pouze pro rozšíření obzoru a znalosti jejich rozdílností.

Poslední z teoretické části je téma Jailbreak. Jedno z nejdiskutovanějších a nejrozsáhlejších témat týkajících se systému iOS. Kvůli jeho zásahu do bezpečnosti systému je pro účel této práce, a vůbec testování aplikací, důležitý.

Druhá polovina této práce se už týká samotné přípravy zařízení. Ta se týká jak telefonu, tak i počítače. K této přípravě patří i předchozí průzkum jiných řešení. Cílem této práce je tedy nikoliv vypracování naprosto originálního postupu, ale zreplikování již hotových řešení a jejich následné zhodnocení.

Již na úvod upozorňuji, že práce nemá za cíl nikoho nabádat k provádění Jailbreaku, i přes to, že některé zdroje uvádějí, že byl uznán legálním, a jiných zásahů do systémů.

I. TEORETICKÁ ČÁST

1 ZÁKLADY TESTOVÁNÍ SOFTWARE

Tato kapitola se zaměřuje na testování softwaru, její hlavní myšlenku a odpovědi na otázky – co to je a proč se to vůbec dělá. Součástí je také historie testování pro doplnění přehledu této problematiky a nebudou chybět ani konkrétní metody testování. Jejich krátký úvod, popis a vysvětlení.

1.1 Co je to testování a proč se dělá

Testování je odhalování chyb v programu. Je důležité myslet na to, že testování nám poví, že program nějakou chybu obsahuje, nikoliv, že je zcela bezchybný! Po samotném programování je nejdůležitější fází vývoje a díky němu jsme schopni dodávat zákazníkovi software, který už od začátku je kvalitnější, než by byl bez testování. Včasné testování dokáže ušetřit spoustu času, peněz a energie.[5][6]

1.2 Krátká historie

Ve světě se začíná řešit debugging a testování už v 50. letech 20. století. Charles L. Baker rozlišuje ve své knize testování od debuggingu. Později vzniká první tým pro testování softwaru při prvním vyslání rakety do vesmíru s člověkem na palubě. V roce 1979 Glenford Myers vydává knihu *The Art of Software Testing*, která se věnuje čistě testování software a udává směr „moderního“ testování. O pár let později, v roce 1984 se uskuteční první konference testování ve Washingtonu, D.C., poslední se konala v roce 2002. Testování jako samostatná, plnohodnotná pozice vzniká až koncem 80. let, u nás je to až 20 let později.[4]

1.3 Testovací metody

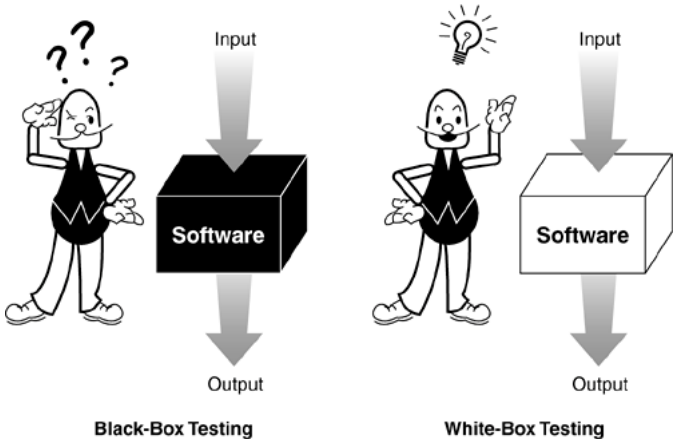
Jako v jiných oborech, i v testování software jsou jisté typy, podle kterých můžeme rozlišit různé postupy testování. V následujících podkapitolách uvedu pouze ty nejčastější a nejvíce zmiňované.

1.3.1 Statické a dynamické testování

Statické testy jsou takové testy, které ke své činnosti nepotřebují běžící program. Především jsou dělány v raných fázích vývoje. Výhodou tak je jejich použití na kontrolu specifikace požadavků a statickou analýzu[1]. Dynamické, na rozdíl od statických testů, potřebují běžící program, anebo jeho funkční část. Jsou dělány spíše v pozdějších fázích vývoje.

1.3.2 Black box a white box

Zde záleží na tom, zda do kódu „vidíme“ - máme k němu přístup, do kódu „nevidíme“, ale víme, jak funguje, nebo nevidíme a ani nevíme, jak funguje.



Obrázek 1: Typy testování

Black box

Jsou to testy, kdy vůbec nevíme, jak je program implementován. Takové testování je z pohledu uživatele, který nemá ponětí o tom, co a jak se uvnitř děje. Pouze víme, jaké jsou vstupy, a jaké následné výstupy. Dá se říct, že je to simulace reálného způsobu použití, a tak člověk, který test provádí, vůbec nemusí být programátor.

[1] Statická analýza = sada metod pro analýzu programů, bez spuštění samotného programu, např. běžně prováděna na nějaké verzi zdrojového kódu.

White box (také Glass box, Transparent box)

U testů, které bychom zařadili do bílé skříňky, máme plný přístup ke kódu programu. Víme tedy v jakékoliv části, co a jak bude dělat a podle toho přizpůsobit test. Zpravidla tyto testy provádí programátor a použití je také typické pro jednotkové testy. Uvádí se i nevýhoda těchto testů, kterou je rozdílnost testovaných scénářů od reálných.

Gray box

Sem zařadíme kombinaci testů černé a bílé skříňky. Jsou navrhovány jako u černé skříňky (z pohledu uživatele), ale můžeme znát implementaci některé části, a tak tento test podle toho upravit a dosáhnout díky tomu přesnějšího testu.

1.3.3 Regresní testování

Pokud jsme na již dříve otestovaném softwaru provedli změny, pak další provedení funkčních a nefunkčních testů pro zjištění, že program stále funguje, nazýváme regresní.

2 ARCHITEKTURA IOS

iOS je mobilní operační systém společnosti Apple. Je to systém na bázi UNIXu a je určen pouze pro mobilní zařízení – telefony iPhone a dříve i pro tablety iPad.[2] První verze s názvem iPhone OS byla vydána spolu s telefonem iPhone v roce 2007 a od té doby oba produkty prodělaly mnoho změn a vylepšení. V současné době je aktuální verze 13.6.

2.1 Funkce iOS

Systém iOS se dělí na 4 vrstvy, které dále obsahují různé frameworky. Seřazeny jsou zde od nejvyšší po nejnižší.

Framework

Pod pojmem Framework si můžeme představit nějakou skupinu knihoven, hlavičkových souborů, multimediálních souborů, které mají pomáhat řešit nejrůznější problémy při programování a tím i ulehčovat a zrychlovat práci programátora.

Springboard

Je to výchozí aplikace, ve které běží veškeré aplikace, interakce atd. Již po odemknutí telefonu se ocitáme právě v prostředí Springboard.

iCloud

Cloudové řešení Applu. Uživatelé na něj mohou ukládat nejen kdejaké soubory, ale i emaily, hesla, kontakty atd. Je zde i možnost automatického zálohování nejdůležitějších dat telefonu.

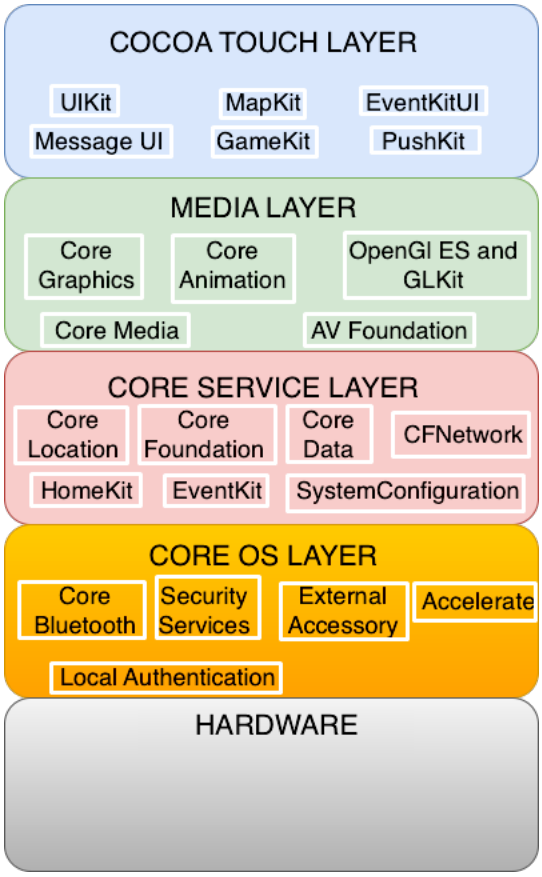
Čas u obrazovky

S příchodem předposlední verze systému iOS byla představena nová funkce Čas u obrazovky. Ta, v případě, že ji má uživatel zapnutou, zaznamenává aktivitu uživatele na zařízení a poslední den v týdnu podá hlášení se statistikami za ten uplynulý týden. Vždy v oznámení sdělí, jak moc se strávený čas na telefonu zvýšil/snížil a v aplikaci ukáže detailnější záznamy.

[2] iPady měly původně systém iOS. Od verze iOS 13 se v Applu rozhodli změnit značení systému pro iPady, nově tedy iPadOS. Je to z důvodu rozšíření iPadů o nové funkce, které jsou určeny pouze pro ně.

2.1.1 Vrstvy systému

Jak název napovídá, systém iOS se skládá z vrstev. Nižší vrstvy poskytují nejzákladnější funkce, na těch vyšších je závislé uživatelské prostředí a grafika. Díky takovému rozložení je zajištěno, že aplikace a hardware spolu nekomunikují přímo.[8]



Obrázek 2: Vrstvy iOS

Cocoa Touch Layer

Na této vrstvě jsou veškeré důležité frameworky pro tvorbu aplikací. Většina jednoduchých, nekomplexních aplikací využívá frameworky převážně z této vrstvy. Jako nejdůležitější Framework této vrstvy bych označil UIKit Framework. Díky UIKitu je systém schopný sestavit a spustit všechny aplikace, které vytvoříme, nebo jen stáhneme. Zajišťuje také funkcionalitu například multitaskingu (= schopnost „souběhu“ více aplikací najednou), push notifikací, nebo i bannerových reklam v aplikacích (reklama, která se zobrazuje pouze na malé části obrazovky, uživatele tak příliš „neotravuje“ jako klasické celo-obrazovkové reklamy a je schopný s takovou reklamou aplikace normálně používat).

Media Layer

Vrstva Media má multimediální funkci. Obsahuje frameworky pro grafiku, zvuky a videa. V případě, že se vývojář rozhodne nepoužít standardní grafiku systému, bude muset využít některých z frameworků, co se věnují grafice.

U této vrstvy je nejzajímavější grafické API Metal. Je hardwarově akcelerovaný, díky tomu dokáže vykreslovat mnohem lepší grafiku s menší náročností na CPU než jeho předchůdce OpenGL.

Core Services Layer

Třetí vrstva shora obsahuje balíčky s nástroji pro obsluhu eventů (akcí) v kalendáři, přístup k databázi kontaktů v telefonu, obsluhu komunikace zařízení s webovou službou iCloud. Také obsluhuje In-App nákupy (= nákupy uvnitř aplikací za nadstandardní výbavy nebo zrušení reklam), komunikace s internetovým obchodem App Store jako takovým ve své oficiální aplikaci, komunikace po síti a informace o mobilní síti.

Mimo obsluhu komunikací také obsahuje framework pro zjišťování a práci s lokací, ukládání uživatelských dat do SQL databáze, nebo i funkcionalitu nové funkce Čas u obrazovky.

Core OS Layer

V poslední vrstvě je už jen pár frameworků. První z nich umožňuje komunikaci se zařízeními připojenými přes Bluetooth LE (Low Energy - nízká spotřeba, patří sem například chytré fitness náramky). Další framework umožňuje komunikaci s hardwarovou vrstvou – přímá komunikace s jednotlivými prvky v těle zařízení.

Za zmínku také stojí frameworky s bezpečnostními prvky pro autentizaci uživatele a autorizaci jeho činností na zařízení, ukládání hesel do „klíčenky“ (zašifrovaná databáze uložených hesel) nebo také obsluhu kryptografických potřeb.

Hardware Layer

Tato vrstva nepotřebuje speciální vysvětlení – patří sem veškeré prvky, které lze najít v těle zařízení.

2.2 Aktuální iOS

Společnost Apple vydává každý rok novou verzi operačního systému pro jejich telefony a mezi nimi vydává „záplatové“ aktualizace. Od své první verze iOS prošel mnoho úpravami. Největší změny proběhly v roce 2013, kdy byla vydána verze iOS 7. Apple zde od základů předělal vzhled a začal používat 64-bit architekturu.

2.2.1 iOS 13

19. září 2019 Apple vydal už 13. verzi operačního systému pro své zařízení iPhone a iPod Touch. Nejvýraznější novou funkcí byl Dark Mode (tmavý režim) pro celý systém. Do té doby bylo možné zapínat dočasně funkci chytrá inverze barev – nástupce pouhé inverze barev. Mezi nimi byl hlavní rozdíl v tom, že chytrá inverze neinvertovala barvy obrázků a fotek. Dark mode ovšem přepíná do tmavého režimu všechny nativní aplikace od Applu a v aplikacích od vývojářů přibývá možnost měnit vzhled na základě nastaveného módu v samotném systému.

Další novinkou v této verzi je Apple Arcade. K tomu je zapotřebí měsíční předplatné v rámci kterého uživatel získá přístup ke hrám, jejichž počet čítá více než sto a stále přibývají.

Mimo tmavý režim a arkádové hry byla vydána nová verze ARKitu, rozšířené reality, která nyní dokáže umisťovat předměty nejen před lidi, ale i za ně.

Poslední novinka, kterou chci zmínit, se týká bezpečnosti. Apple spustil službu Sign in with Apple. Můžeme se s tím setkat například na některých webech, nebo v některých aplikacích. Funguje tak, že namísto vyplňování osobních údajů použije nás účet Apple ID a bude poskytovat jen email, případně je zde možnost využít náhodně vygenerované adresy, přes kterou nám bude Apple přeposílat zprávy, ale nebude přitom nikde veřejně použita naše osobní emailová adresa.

2.3 Rozdíly systémů Android a iOS, jejich funkcionalita

Na trhu s mobilními zařízeními ovšem není pouze Apple se svými iPhony. Hlavním rivalem je zde Google s operačním systémem Android. Oba tyto systémy plní kromě základních funkcí telefonu (SMS, hovory a v posledních letech i mobilní internet) i nespočet dalších užitečných možností a dost se od sebe liší. Prvního rozdílu je možné si všimnout už při pohledu na zařízení s těmito systémy - tím je design. Odlišné zamykací obrazovky, uspořádání plochy, struktura a stavba aplikací. I přesto, že oba systémy prošly zásadní změnou vzhledu. V případě Applu to bylo ve verzi iOS 7, kdy přešel na flat design z tzv. **skeuomorfismu**. Používali imitace reálných objektů a materiálů v designu aplikací, například ikona aplikace poznámek byla z části kožená a z druhé připomínala klasický nažloutlý papír. Flat design to značně zjednodušuje, vzhled je minimalističtější. U Androidu to bylo ve verzi 5 Lollipop, kde přešli na Material design, který je podobný Flat designu.

Oba systému disponují také hlasovými asistenty. Apple přišel se Siri, na druhé straně přišli s Google Now. Tito asistenti jsou schopni rozeznat širokou škálu příkazů, s trochou štěstí se s nimi dá i popovídat, ovšem lidskou komunikaci to zdaleka nenahradí. Některé zdroje uvádějí, že Google Now je propracovanější a tak toho zvládne více.

Zatímco iOS je tvořen a dodáván pouze na telefonech iPhone, či multimediálními zařízeními iPod, Android je open source a tak je možné se s ním setkat na telefonech různých značek. Díky tomu zde hapruje plynulost na některých výkonem méně obdařených modelech. Tak jako má iOS verze tvořené přímo na konkrétní modely, a tak je na nich běh plynulý i po delší době od vydání aktualizace a uživatelé jsou schopni s telefonech fungovat poměrně dlouhou dobu, Android je tvořen primárně na telefony od Googlu, značky Nexus. Uživatelé těchto telefonů mají aktualizace nejdřív a jsou také nejméně problémové.

Pro nás je ale nejzásadnějším rozdílem struktura systému. iOS je poměrně uzavřený systém. Běžný uživatel nemá šanci pozměnit věci, které mu Apple vyloženě nedovolí. Není ani oprávněn nahlédnout do systémových souborů, má přístup pouze k datům, které se vážou ke konkrétní aplikaci. Díky tomu je systém slušně chráněn proti nechtěným zásahům do jeho souborů a nejrůznějším virům. Android je na tom zcela opačně, uživatelé mají plný přístup k souborům systému a je velmi náchylný na napadení viry.

Za zmínku stojí také jejich obchody s aplikacemi. Veškeré nové aplikace na AppStore Apple kontroluje a pokud aplikace nesplňuje přísné požadavky, je nebezpečná, Apple ji neschválí a na standardní iOS se nedostane. Naopak na Google Play je možné sehnat různé aplikace klidně obsahující kdejaké viry, zde si tedy musí dát uživatelé velký pozor na to, co stahují.
Posledním bodem bych chtěl zmínit Jailbreak, o kterém je ale celá následující kapitola. V dřívějších verzích iOS nebylo tolik nástrojů a volnosti co se týče úprav vzhledu apod. jako teď. Proto lidé hojně vyhledávali způsoby, jak to obejít. Proto vznikl Jailbreak. Uživatelé, co si jej na telefonu udělali, nebo nechali udělat tehdy ještě i například za peníze cizími lidmi, si takto otevřeli systém. Mohli si jej upravit, jak jen chtěli, přidat různé nástroje, ale také na telefon instalovat cracklé hry a aplikace. Narušili si tím bezpečnost systému, odemkli přístup k systémovým souborům, a byla tedy vysoká šance, že si do telefonu nainstalují nějaký škodlivý software. Android už od základu je dosti otevřený, dovoluje nejrůznější úpravy a instalaci neověřených, cracklých aplikací. Nepotřebuje tedy Jailbreak, ale je zde možné provést root telefonu. Z uživatele se tím stane „naduživatel“ a je mu dovoleno už naprosto vše, co na telefonu dělat jde.[9][10]

Tabulka 1: Rozdíly mezi iOS a Android

Rozdíly	iOS	Android
Design	Flat	Material
Hlasový asistent	Siri	Google Now
Dostupné na	Pouze iPhone, iPod	Jakákoliv značka
Systém	Poměrně uzavřený	Otevřený
Obchod s aplikacemi	AppStore	Google Play
Vývojové prostředí	Xcode	Android Studio
Programovací jazyk	Swift	Java

3 JAILBREAK

Jailbreak, jak název napovídá, znamená jakýsi „únik“ z cely, kterou je iOS. Apple nám v mobilním systému nenechává moc volnosti, co se týče možností, jak si věci upravit, změnit, zakázat. Je to velmi uzavřený systém a běžný uživatel nemá šanci dostat se například do systémových souborů, nebo si přidat nějakou užitečnou funkci, kterou tam Apple nepřidal.

Není zde také možnost nainstalovat aplikaci, která není k sehnání v oficiální obchodu App Store.

Právě k obejití těchto „nemožností“ nám pomůže Jailbreak. I když se zdá, že Jailbreak má samé výhody, tak tomu tak opravdu není. Díky tomu, že nám systém „otevře“, zde vzniká velké riziko, že se do našeho zařízení může dostat vir. Může nám používání iPhonu znepříjemnit, nebo i ukrást citlivá data jako mohou být osobní údaje, fotografie, či bankovní údaje.

3.1 Historie Jailbreaku

Jailbreak vznikl už v roce 2007, velmi krátce po vydání vůbec prvního telefonu iPhone. Uživatelé si díky tomu například mohli do telefonu nahrát aplikace a mít tam tedy i jiné než jen ty oficiální, které byly natvrdo v systému. V té době totiž neexistoval obchod s aplikacemi App Store. S příchodem každé nové verze systému se na internetu objevovaly i nové verze jailbreaku. Hlavními důvody, proč to lidé chtěli a dělali, byly následující:

- Chtěli placené aplikace zdarma,
- Chtěli si přizpůsobit vzhled systému tak, jak to Apple nedovoloval,
- Chtěli si do systému přidat nástroje, které tam nebyly a měli je konkurenti.

Applu trvalo pár let, než začal plnit přání uživatelů a s každou novou verzí systému přichází novinka, kterou si lidé museli přidávat do svých telefonů jen díky jailbreaku. Díky tomu už pomalu Jailbreak ztrácí své kouzlo a význam, a tak se snižuje procento lidí, kteří si jej na svých zařízeních provedou. Apple se také snaží těmto možnostem zabránit, takže s každou novou verzí systému nepřichází jen nějaká novinka, je čím dál těžší najít takové exploity, které by Jailbreak umožnily.

3.2 Typy

Stejně jako jinde, i Jailbreak má různé typy instalace. Za dobu své existence se dařilo jeho vývojářům instalovat jej v celkem 4 různých možnostech. [14][15]

3.2.1 Tethered

Tento typ je nejhorší volbou jailbreaku. Je potřeba jej provést přes počítač s připojením přes USB kabel. Takto upravené zařízení následně samo nezvládne návrat do funkčního stavu, myšleno po restartu či z vypnutého stavu. Je třeba jej opět připojit k počítači a přes pomocný software znovu nastartovat.

3.2.2 Untethered

Opak k volbě tethered je untethered. Nejideálnější možnost jailbreaku. Je možné jej nainstalovat jak přes kabel a počítač, tak i na samotném zařízení přes webový prohlížeč Safari. Zařízení s tímto typem je možné restartovat jak je libo, Jailbreak na něm zůstane.

3.2.3 Semi-Tethered

Tento typ vychází z tethered jailbreaku, jen s malým rozdílem. Semi-tethered zvládne po restartu či vypnutí zařízení nastartovat, ovšem bez vylepšení, které tam jsou díky jailbreaku. Díky tomu je uživatel schopen při nechtěnému restartu zařízení alespoň textovat nebo telefonovat, zkrátka používat telefon jako před provedením jailbreaku. Pro používání svých tweaků bude ale potřebovat zase počítač.

3.2.4 Semi-Untethered

Velká podobnost se semi-tethered. Telefon je schopný nastartovat do normálního režimu bez jailbreaku, pokud si ale uživatel chce znovu užívat tweaků a dalších vymožeností, musí si to aktivovat přes aplikaci v telefonu. I tak stále nebude na telefonu plnohodnotný Jailbreak a bude třeba jej do toho stavu přivést počítačem.

3.3 Aktuální stav Jailbreaku

Aktuálně je vydána nejnovější verze iOS 13.4. Byl vydán 24. března 2020 a už je možnost na něm provést Jailbreak. Na internetu je velký počet jailbreaků nejrůznějších tvůrců, ovšem žádný z nich není permanentní, je typu semi-tethered. Takovou chybu v systému, která by to umožňovala, se zatím nikomu nalézt nepodařilo.

II. PRAKTICKÁ ČÁST

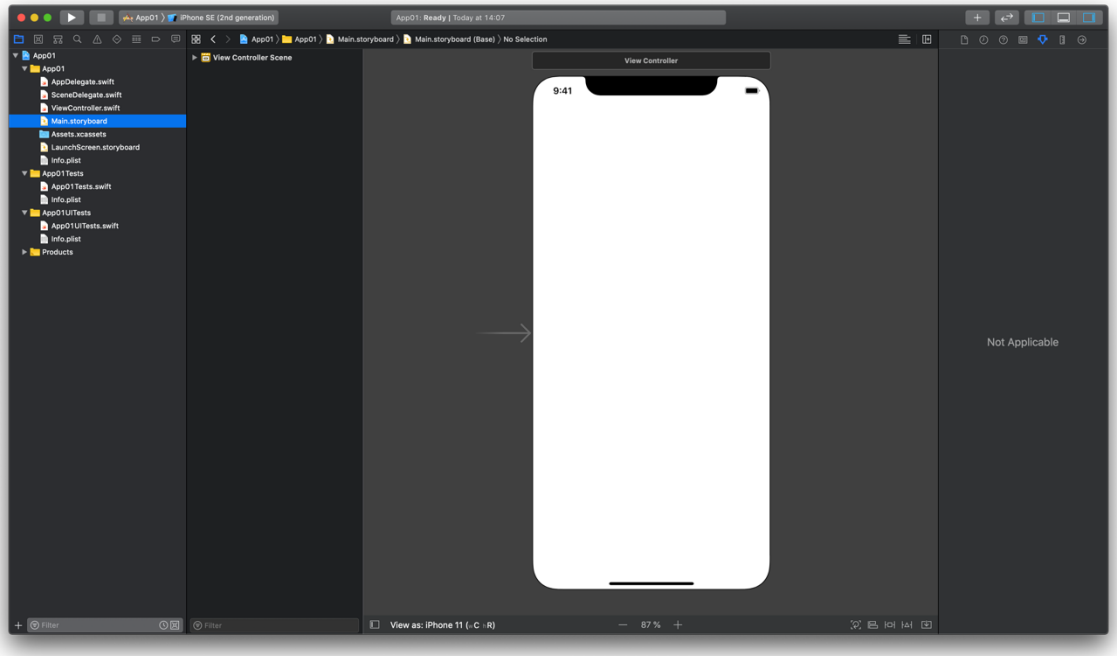
4 TESTOVACÍ PROSTŘEDÍ

Pro nativní vývoj na systémy iOS, macOS atd. je zapotřebí Mac/Macbook se systémem macOS a vývojové prostředí Xcode, které je rovněž dílem Applu.

V případě telefonu je v práci využíván iPhone 7 Plus, s verzí systému iOS 13.5.1. Dále je použit MacBook Pro 2017 s verzí macOS 10.15.5.

4.1 Xcode

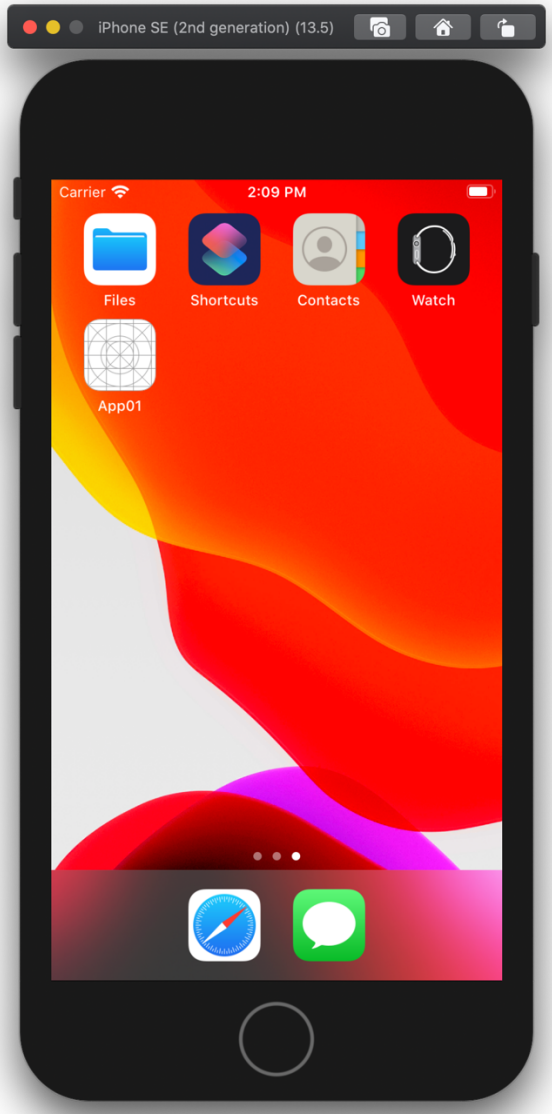
Vývojové prostředí z dílen společnosti Apple. Tvořeno pro vývoj aplikací pro všechna jablečná zařízení – od hodinek Apple Watch, přes telefony iPhone a tablety iPad po počítače MacBook a Mac. Na internetu je tedy plná podpora pro takový vývoj, i široká škála dokumentací, řešených problémů a jejich řešení.



Obrázek 3: Xcode

4.1.1 Emulátory

Apple dodává v prostředí Xcode i jejich vlastní emulátor iPhonů, iPodů a iPadů. Má název Simulator a svou dokonalostí dokáže z velké části nahradit fyzické zařízení. Vývojář je schopen kvalitního vývoje i bez reálného telefonu.



Obrázek 4: Simulator

4.2 Příprava telefonu

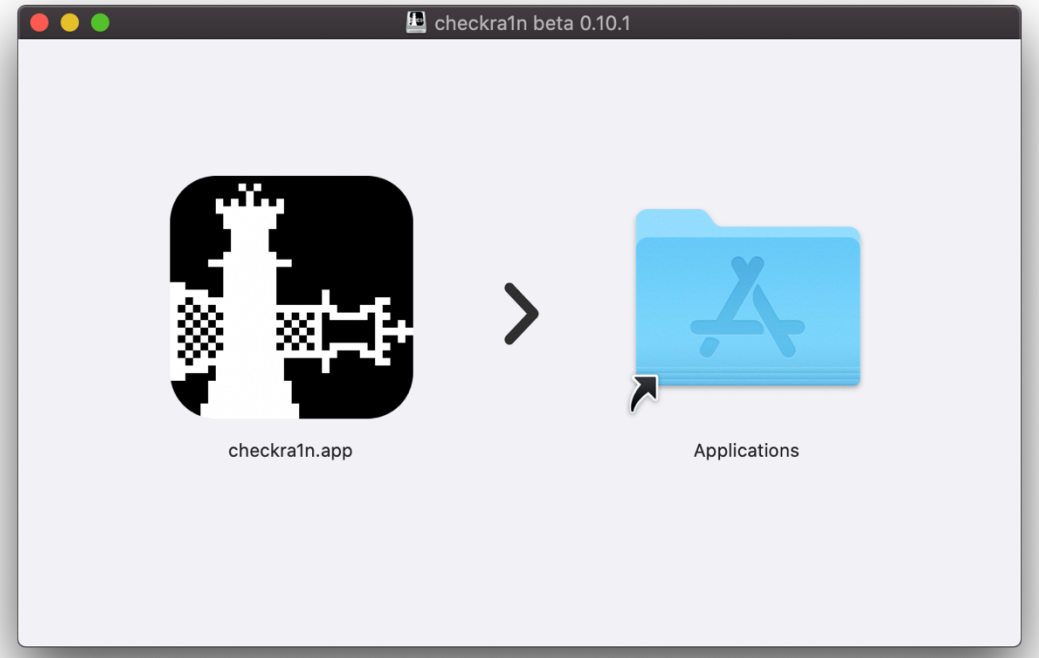
Jednodušší a rychlejší část přípravy potřebných zařízení se týká telefonu. Na něj je nezbytné nainstalovat Jailbreak a díky tomu nainstalovat pár nástrojů.

4.2.1 Jailbreak

Víme, že verzí jailbreaku pro poslední vydaný iOS je více. V této kapitole se bude používat *checkra1n.* Pro lehké provedené vývojáři tohoto jailbreaku vyvinuli i aplikaci, ta je ovšem dostupná jen pro počítače Mac a počítače se systémem Linux. Zařízení používaná v této kapitole jsou mezi podporovanými zařízeními na oficiálních stránkách tvůrců *checkra1nu*.[7]

Instalace aplikace

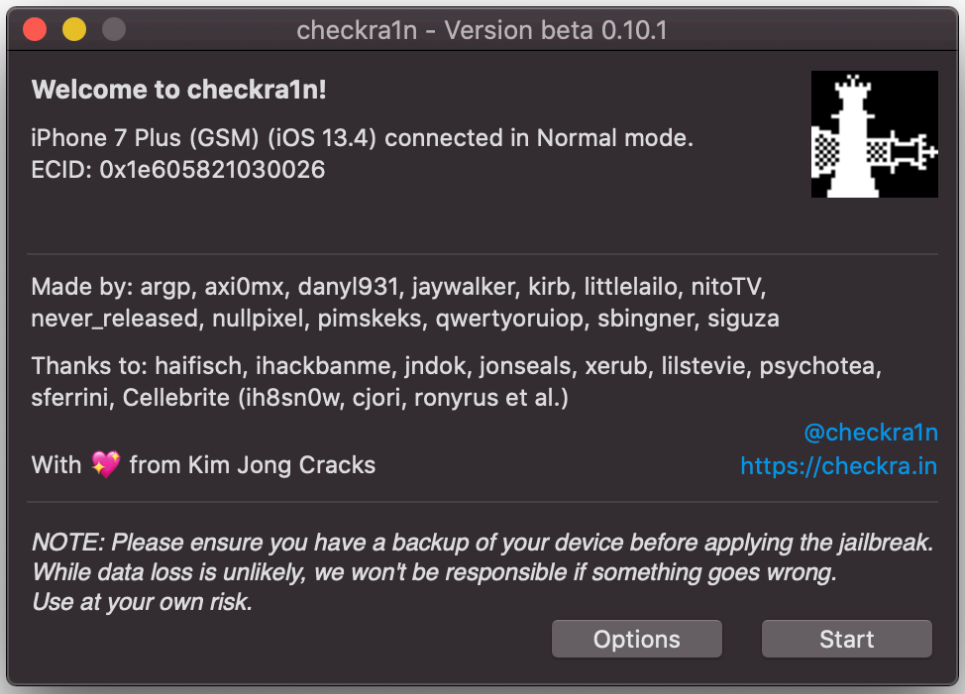
Po stažení a otevření instalátoru je vyžadován klasický krok pro instalaci, jako u většiny aplikací, které nepochází z App Storu – přetažení aplikace do složky Aplikace. V případě, že má uživatel zapnuté zabezpečení FileVault, je naprosto nutné jej vypnout!



Obrázek 5: Instalace checkra1n

Následně je možné spustit aplikaci checkra1n.app. Je ovšem velmi pravděpodobné, že systém MacOS nedovolí spuštění této aplikace. Je tedy nutné ji v předvolbách systému v záložce Zabezpečení a soukromí dodatečně povolit.

Nyní je možné aplikaci spustit. Pokud je vše, jak má, na úvodní obrazovce aplikace je v horní části okna vypsán model iPhonu, který je k počítači připojen.

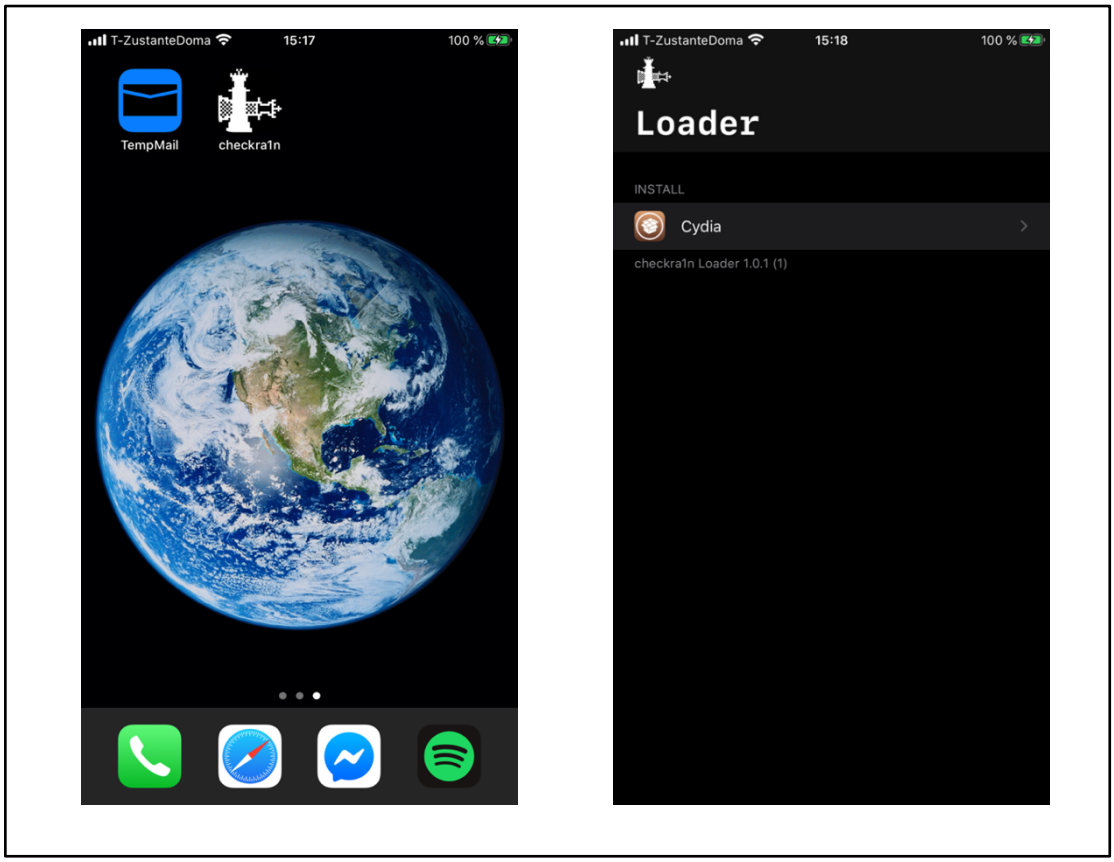


Obrázek 6: Spuštění nástroje checkra1n

Po kliknutí na tlačítko Start se v okně objeví upozornění, že je třeba dostat telefon do DFU módu. To je speciální stav telefonu, kdy zařízení neběží, a lze na něj nainstalovat i jinou (klidně upravenou) verzi systému, než která tam aktuálně je.[11] Aplikace obsahuje nádherně zpracovaný návod, díky kterému tento krok zvládne každý uživatel.

Instalace Cydie

Jakmile se telefon dostane do tohoto stavu, aplikace do něj „vloží“ aplikaci „checkra1n. Tu je potřeba po restartu telefonu spustit a v ní potvrdit instalaci aplikace Cydia. To je výchozí a dá se říct hlavní aplikace pro Jailbreak. Přes ni je teprve možné instalovat veškeré tweaky a upravovat si systém v telefonu.



Obrázek 7: Aplikace checkra1n (vlevo) a instalace aplikace Cydia

V tuto chvíli je Jailbreak telefonu hotový. Dalším krokem je stažení nástrojů v aplikaci Cydia.

4.2.2 Cydia tweaky

V rámci přípravy telefonu nestačí pouhý Jailbreak. Ten zaručí jen „otevření“ systému. Je tu proto seznam pár tweaků[3] (a k některým i výpis věcí, co obsahují), které je potřeba nainstalovat v aplikaci Cydia.

- **Network Commands**
  - arp, ifconfig, route, traceroute
- **Developer -cmds**
  - ctags, haxdump, rpcgen, unifdef
- **Erica Utilities**
- **File-cmds**

[3] Tweak je nástroj (v některých případech i sada nástrojů), který usnadňuje určitou práci, nebo přidává nové možnosti do funkčního systému. Alternativní název je utilita.

    - Chflags, compress, ipcrm, ipcs, pax
- **Gawk**
- **IOKit Tools**
    - IOAllocCount, IOClassCount, IOReg
- **make**
- **nano**
- **unrar**
- **unzip**
- **.syslog commandline**
- **Zip**
- **OpenSSH**
- **OpenSSL**
- **Git**
- **PreferenceLoader**
- **Applist**
- **Clang-10**

Všechny tyto tweaky je možné bez problému nainstalovat na telefon.

4.3 Příprava počítače

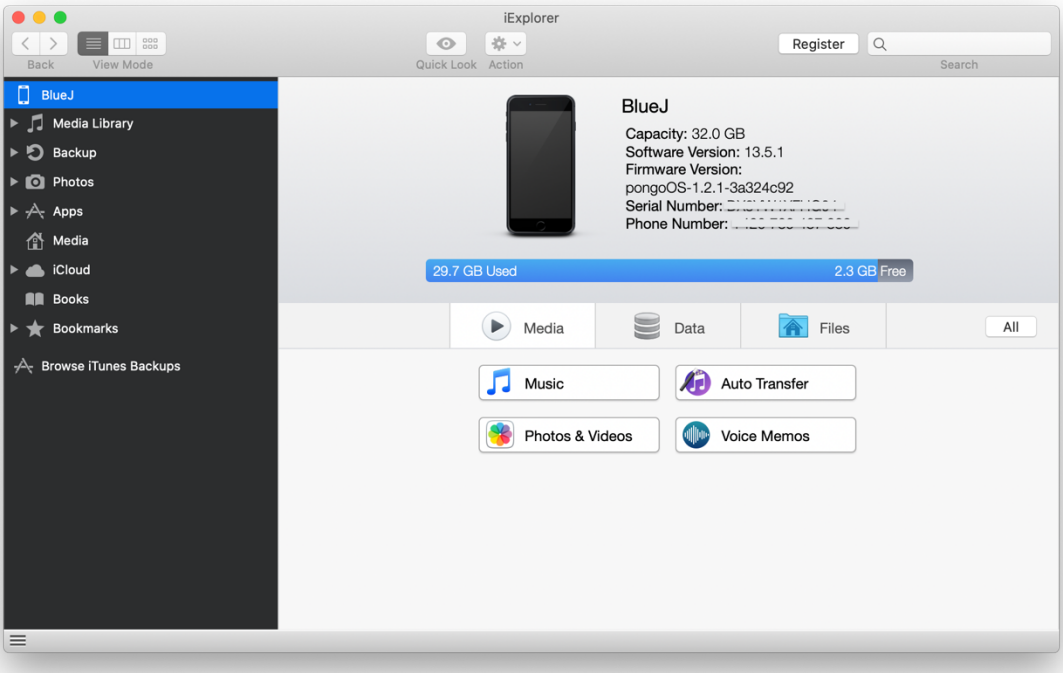
Stejně jako u telefonu, je potřeba mít i speciálně připraven počítač. Pro testování existuje spousta programů a aplikací, ale v této práci ukážeme jen pár z nich.

4.3.1 Instalace programů

Jedna z částí přípravy počítače je jednoduchá instalace aplikací. Není třeba speciálních dovedností a znalostí.

4.3.1.1 iExplorer

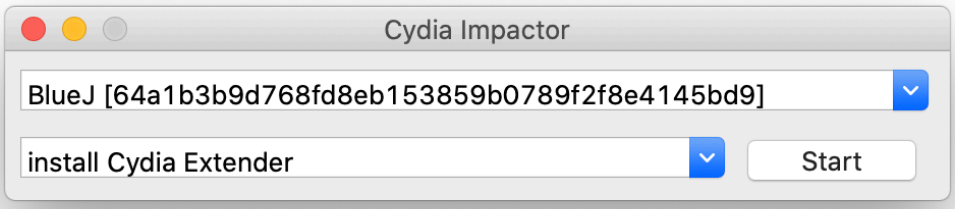
iExplorer je univerzální aplikace pro správu jablečných mobilních zařízení. Oproti klasickému procházení v nativním Finderu na Macu iExplorer umožňuje přehledně procházet fotky, nainstalované aplikace, nahrávky v diktafonu, hudbu a podobně. Je zde také možnost spravování záloh.



Obrázek 8: iExplorer

4.3.1.2 Cydia Impactor

Tato aplikace slouží k instalování aplikací třetích stran do telefonu mimo oficiální AppStore. Vyžaduje k tomu pouze Apple účet (doporučuje se použít jiný, než osobní).



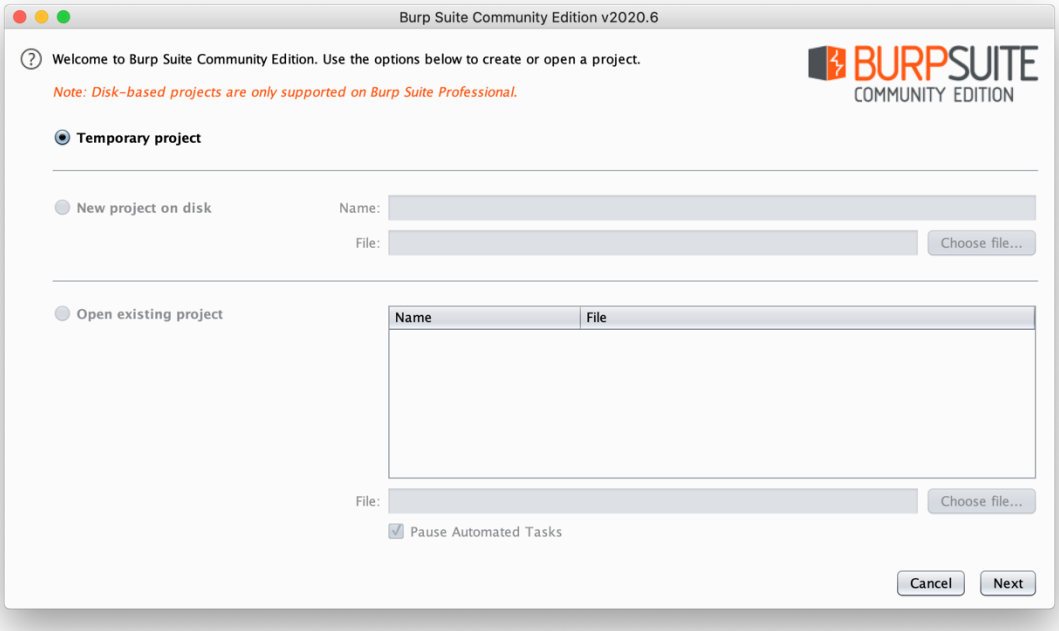
Obrázek 9: Cydia Impactor

4.3.1.3 Burp Suite

Mnoho bezpečnostních profesionálů využívá i různé aplikace pro testování bezpečnosti webů a vůbec komunikace přes internet. Burp je taková aplikace. Díky ní je uživatel schopen zachytávat a filtrovat přenos po síti, požadavky a odpovědi serverů atd. Obsahuje také další užitečné nástroje, například pro dešifrování a opakování požadavků.

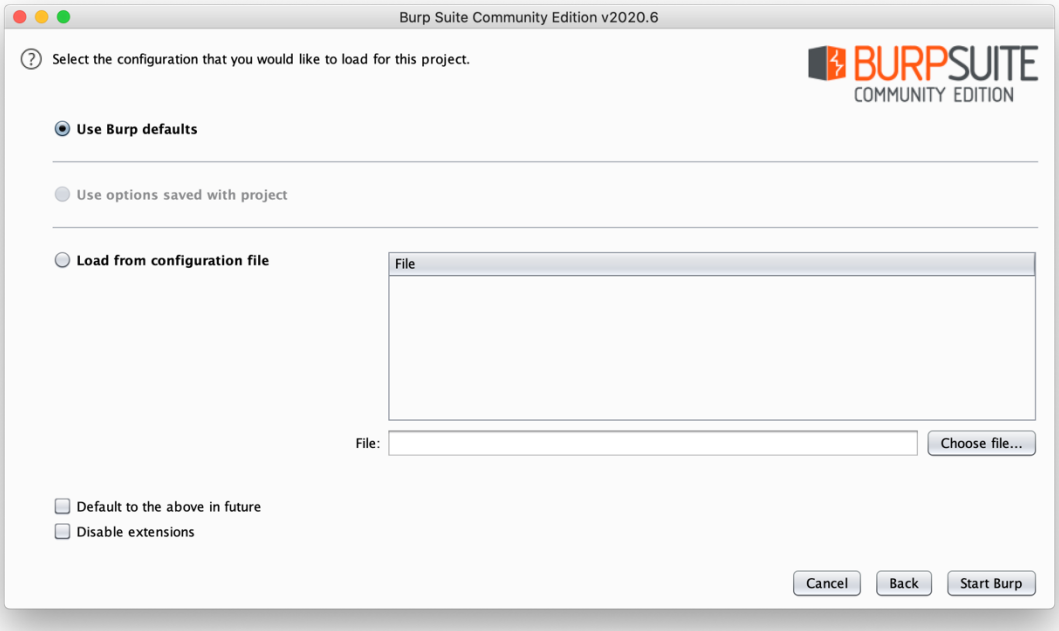
Po spuštění aplikace

Po startu je třeba se „proklikat“ pár okny. V prvním okně Burp dává možnost zvolit si již existující projekt, vytvořit nový, čistý projekt, nebo projekt dočasný. To znamená, že veškerá práce nebude uložena a po ukončení aplikace se ztratí. A právě tato možnost je dostačující.



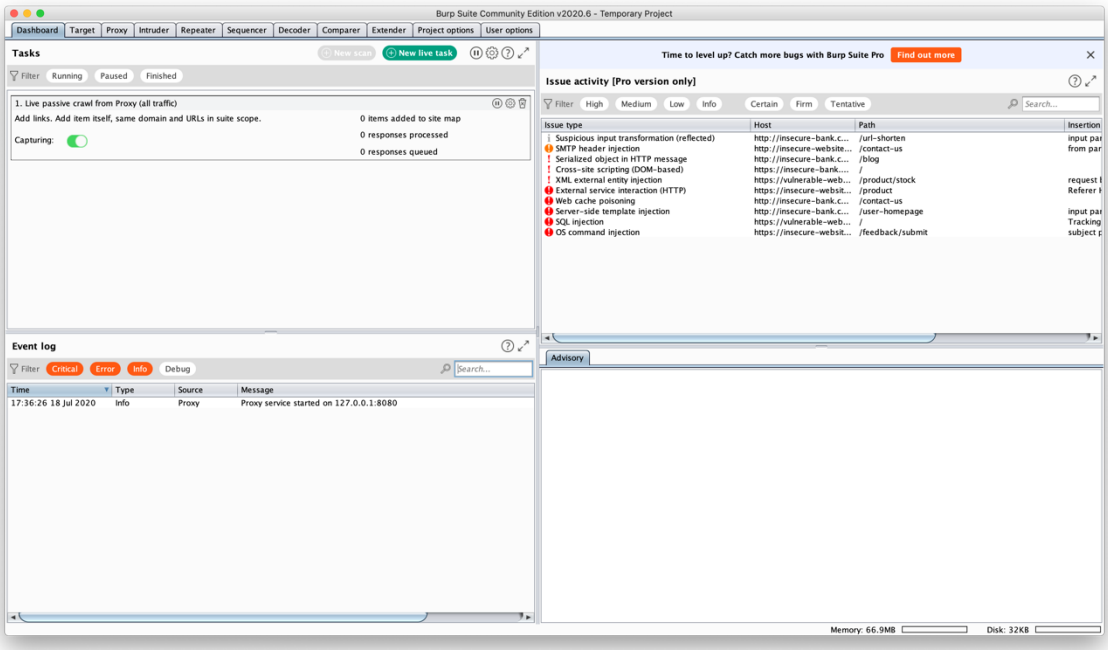
Obrázek 10: Burp Suite - volba projektu

Druhé okno se týká konfiguračního souboru. Je zde možnost buď použít takový soubor v již existujícím projektu (za předpokladu, že v předchozím okně byla volba použití již existujícího projektu), použít samostatný konfigurační soubor uložený mimo aktuální projekt, nebo použít základní konfiguraci. Opět poslední možnost je dostačující pro tento projekt.



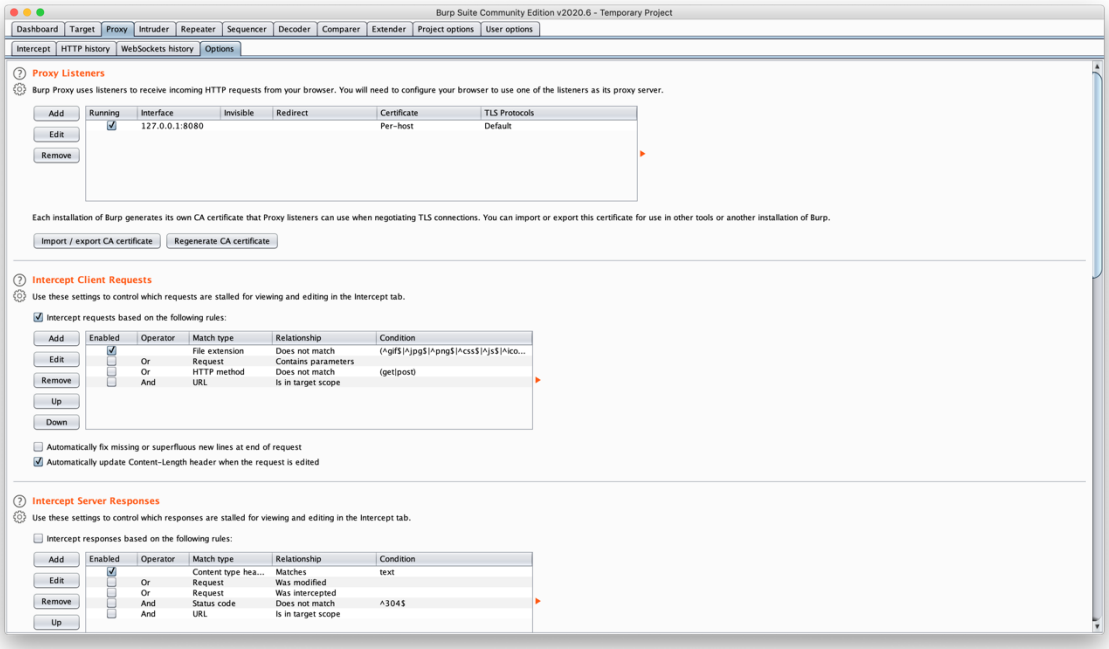
Obrázek 11: Burp Suite - volba konfig. souboru

Hlavní obrazovka je rozcestím pro veškeré nástroje a možnosti aplikace. Je zde už pár oken s informacemi, pro tuto práci je důležité levá strana. Aktivní proces proxy sledovače a event log.



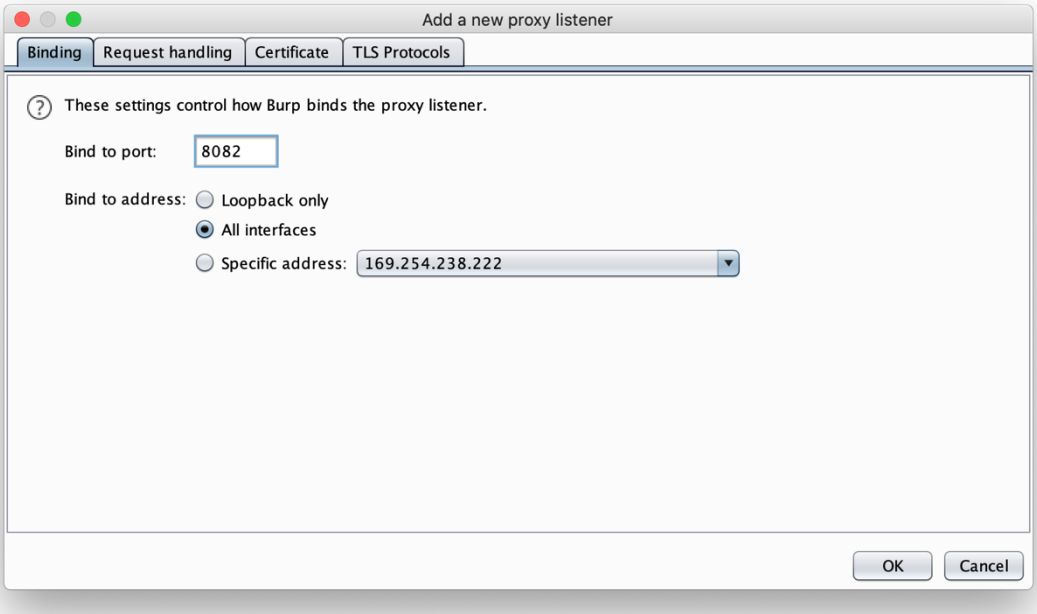
Obrázek 12: Burp Suite - Hlavní okno

V horní části je mnoho záložek, ale důležitá je záložka „Proxy“.



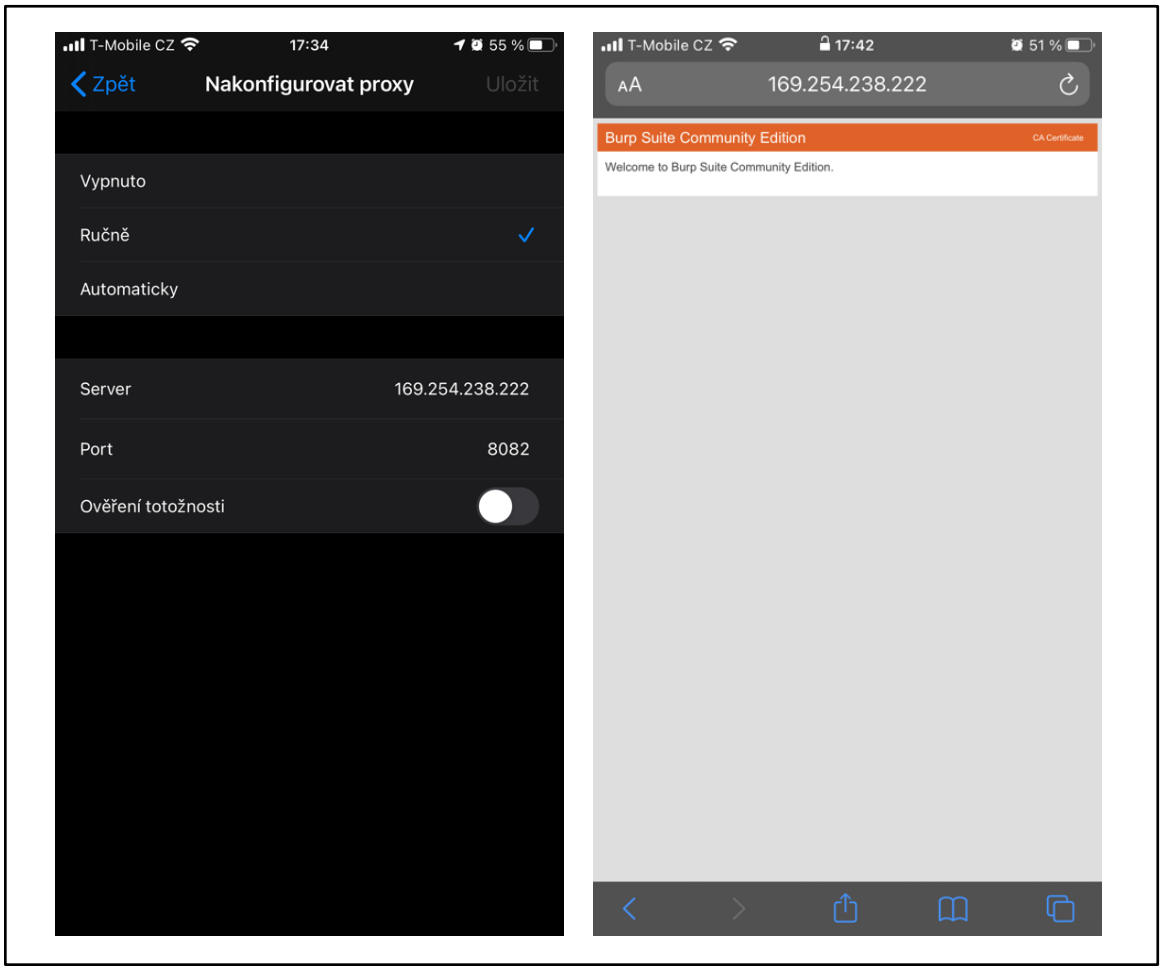
Obrázek 13: Burp Suite - Záložka proxy

Zde je potřeba nastavit zachytávání na všech interfacech (všechny adresy) s portem 8082.



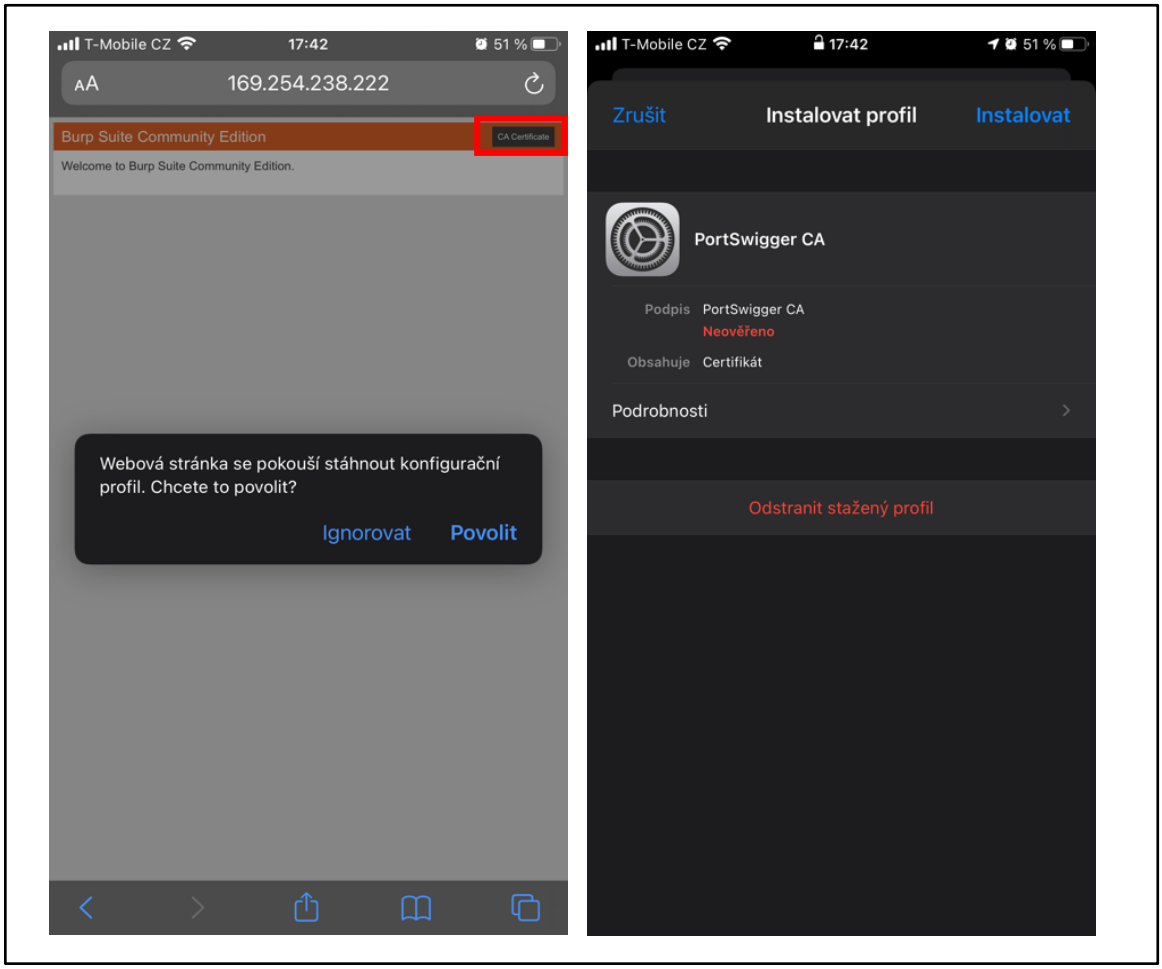
Obrázek 14: Burp Suite - Nastavení proxy

Aby to ovšem pracovalo správně, je potřeba mít zařízení připojeno na stejné síti (nejideálnější je možnost ad-hoc). Dále nakonfigurovat proxy na zařízení.



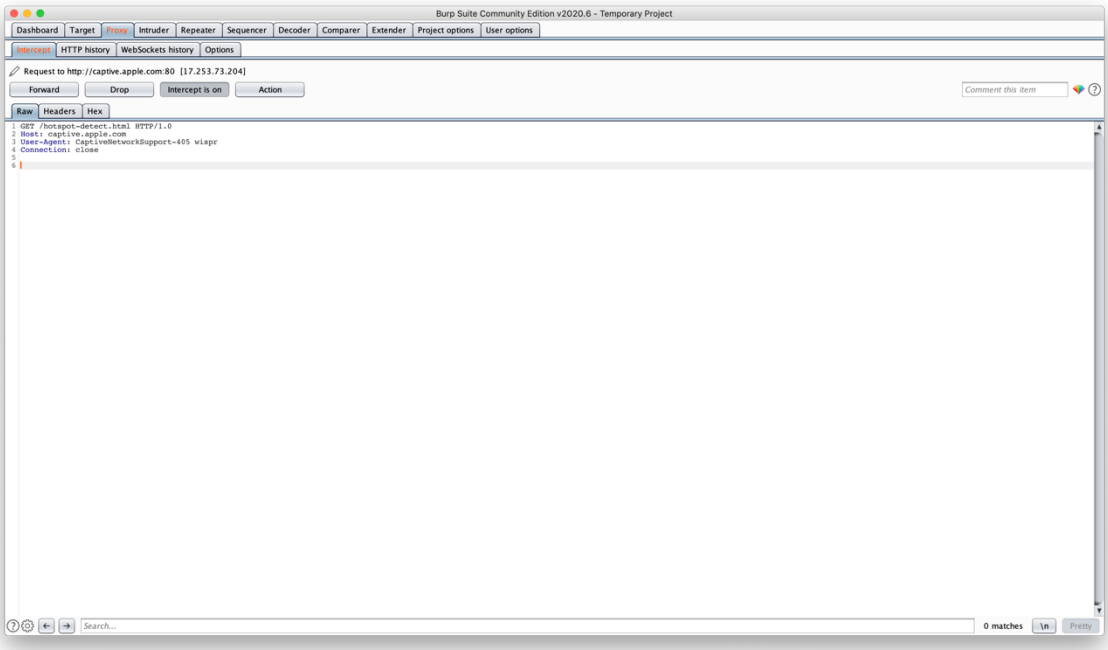
Obrázek 15: konfigurace proxy na telefonu (vlevo) a ověření připojení k Burp Suite

A posledním krokem v přípravě je stažení a instalace konfiguračního profilu s certifikátem. Certifikát lze najít na adrese **http://ip:port**, v tomto případě **http://169.254.238.222:8082**.



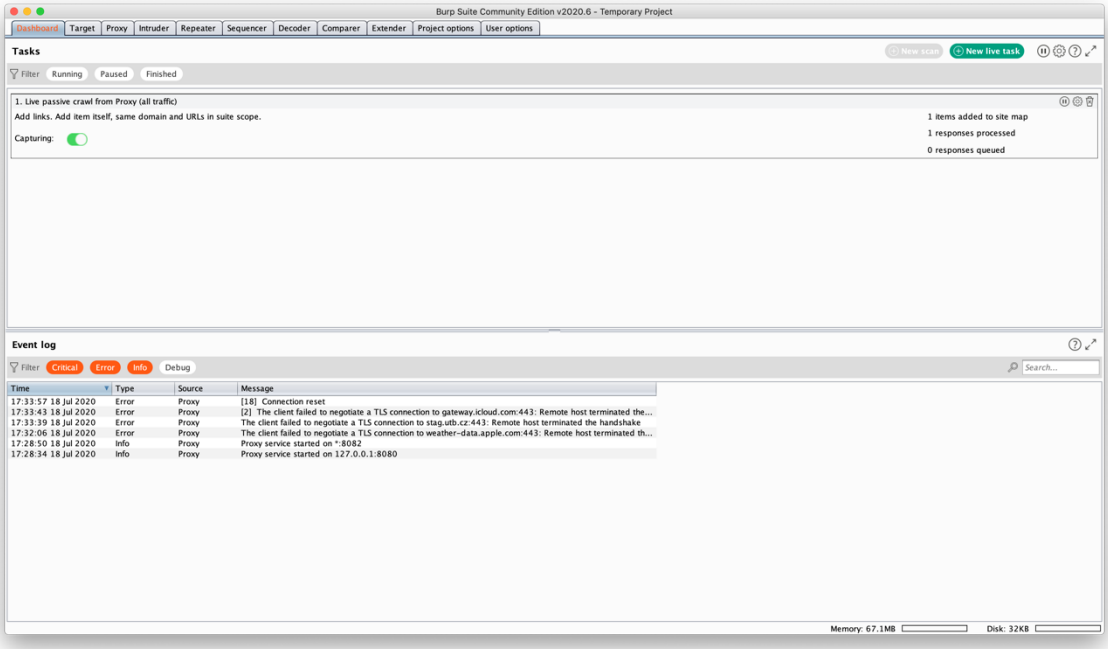
Obrázek 16: Burp Suite - Stažení certifikátu (vlevo) a instalace na iOS

Po úspěšné přípravě telefonu bude Burp zachytávat veškeré požadavky a vypisovat jejich těla.



Obrázek 17: Burp Suite - Odposlouchávání proxy

Je možné vidět historii v event logu na hlavní stránce.



Obrázek 18: Burp Suite - Event log

4.3.2 Externí nástroje pro iOS

Jedním z možných externích nástrojů je **Secure Shell** – neboli SSH. Jedná se o zabezpečený komunikační protokol používaný v rámci sítí užívající TCP/IP. Zajišťuje tedy bezpečnou komunikaci, například v oblasti Internetu a je mnohem bezpečnější, než telnet. V našem případě přes něj budeme komunikovat s telefonem.

Je potřeba zjistit IP adresu telefonu.

- *Nastavení → Wi-Fi → tlačítko „informace“ u připojené sítě → zde vyčíst IP*

Nyní je možné připojit se k telefonu, jsou zde 2 možnosti:

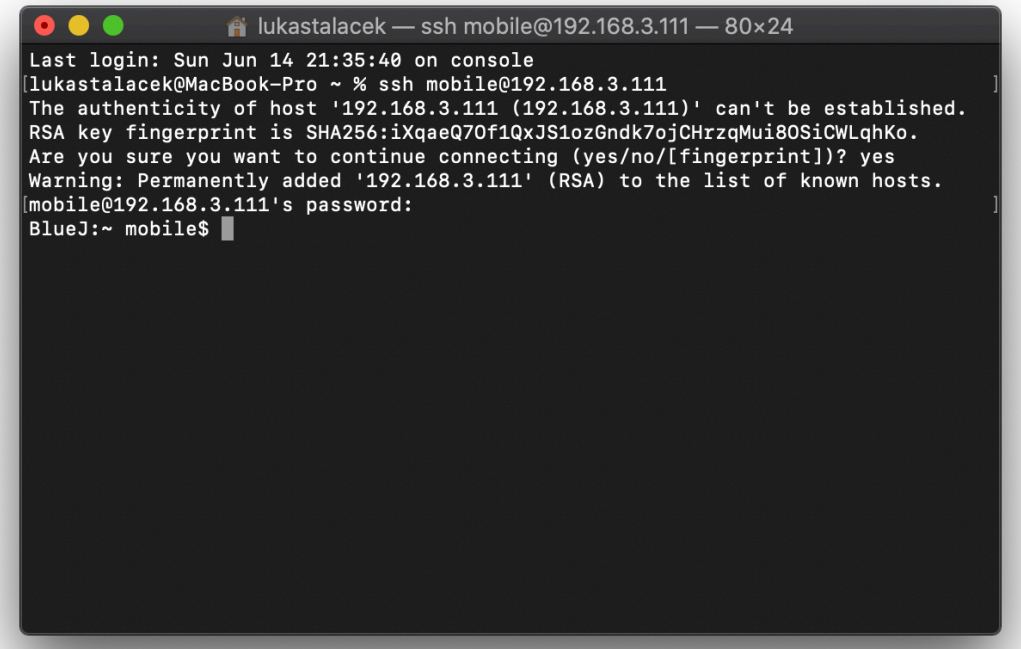
- Přihlásit se jako běžný uživatel - **mobile**
- Přihlásit se jako „naduživatel“ – **root**

U obou přihlášení je prvotní heslo stejné → *„alpine“* a je velmi doporučeno si jej změnit.

Připojení k telefonu proběhne použitím příkazu:

```
⇒ ssh mobile@192.168.3.111
```

a následným zadáním hesla.



Obrázek 19: SSH

Instalace Class-Dump

Tento nástroj není podporován od verze **iOS 11,** proto jeho instalace na aktuální verzi není možná.

Instalace Clutch

Vysokorychlostní dešifrovací nástroj, vyžaduje jailbreak a podporuje všechny verze od iOS 8.

Požadavky pro instalaci jsou:

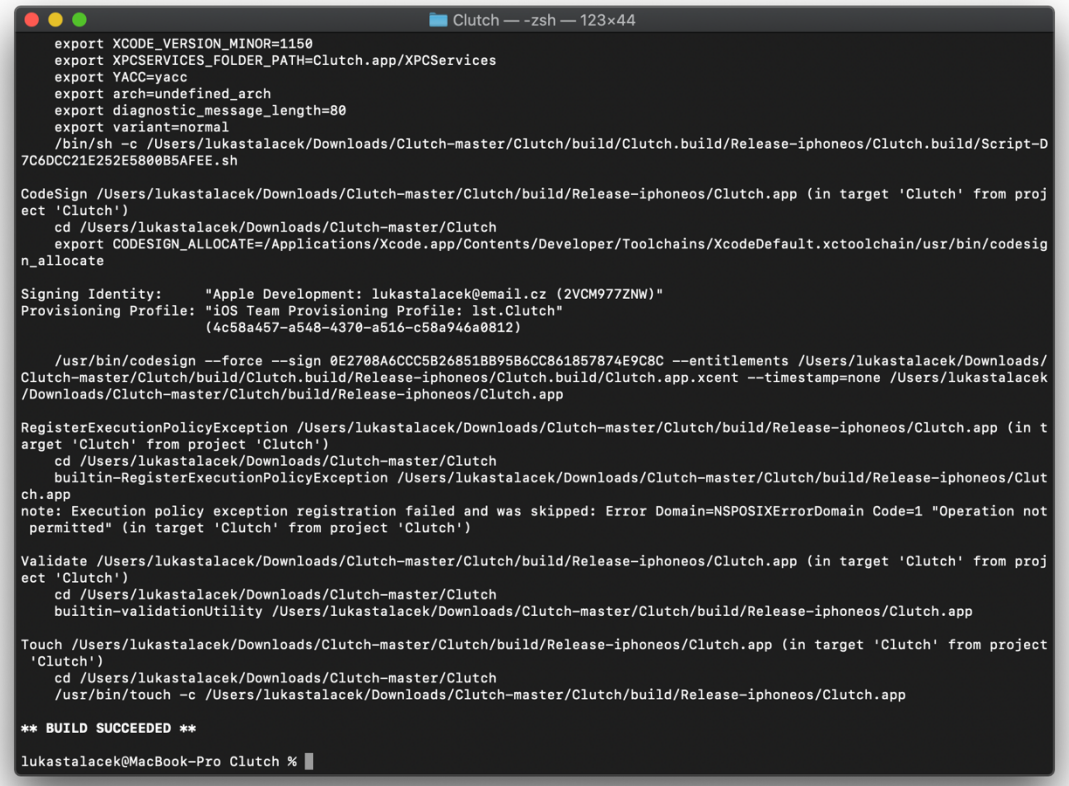
- **Xcode**
- **Xcode nástroje příkazového řádku** (lze stáhnout na oficiálních stránkách, nebo použitím příkazu:

```
#1   xcode-select –install
#2   killall Xcode
#3   cp
     /Applications/Xcode.app/Contents/Developer/Platforms/iPhon
     eOS.platform/Developer/SDKs/iPhoneOS.sdk/SDKSettings.plist
     ~/
#4   sudo /usr/libexec/PlistBuddy -c "Set
     :DefaultProperties:CODE_SIGNING_REQUIRED NO"
     /Applications/Xcode.app/Contents/Developer/Platforms/iPhon
     eOS.platform/Developer/SDKs/iPhoneOS.sdk/SDKSettings.plist
#5   sudo /usr/libexec/PlistBuddy -c "Set
     :DefaultProperties:AD_HOC_CODE_SIGNING_ALLOWED YES"
     /Applications/Xcode.app/Contents/Developer/Platforms/iPhon
     eOS.platform/Developer/SDKs/iPhoneOS.sdk/SDKSettings.plist
```

Je také potřeba vypnout vyžadování podepsaného softwaru. Toho lze dosáhnout přikazy na řádcích 2-5.

Dále je třeba stáhnout z repozitáře soubory do počítače, v terminálu jít do složky „Clutch“ a zde provést příkaz pro sestavení:

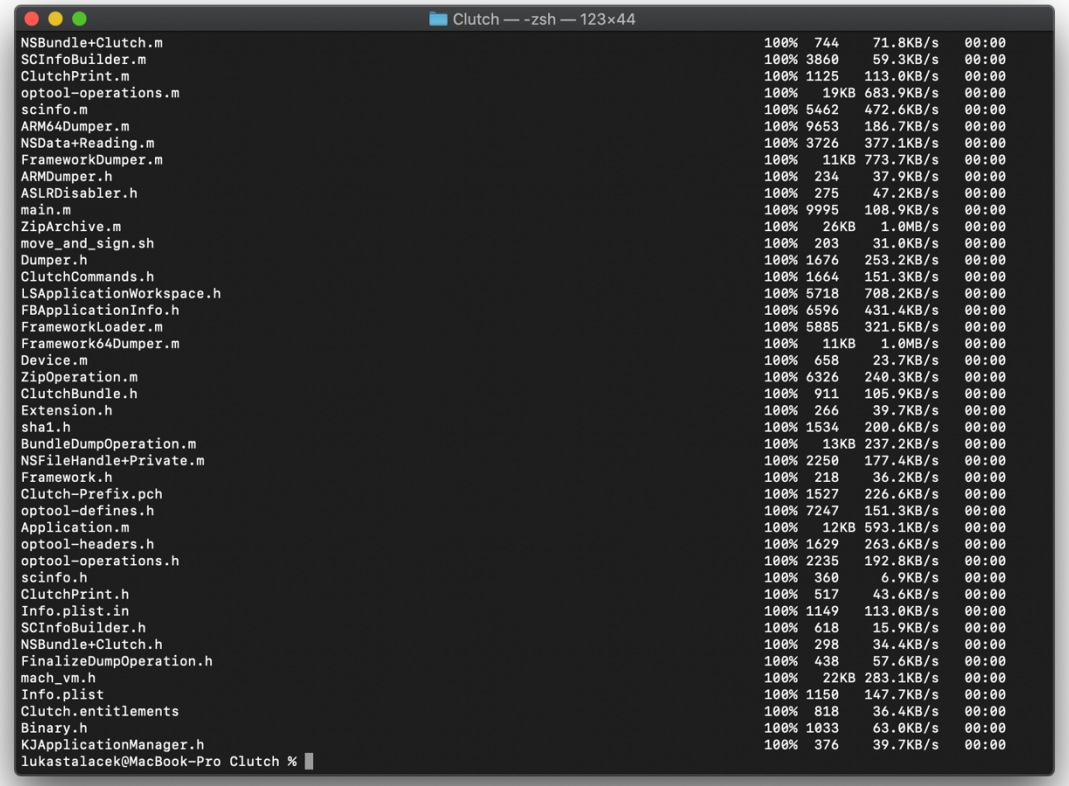
```
⇒ xcodebuild clean build
```



Obrázek 20: Clutch – build

Stále ve stejné složce provést příkaz pro překopírování sestavené aplikace do telefonu:

```
⇒ scp -r Clutch root@192.168.3.111:~
```



Obrázek 21: Clutch

Instalace Introspy iOS

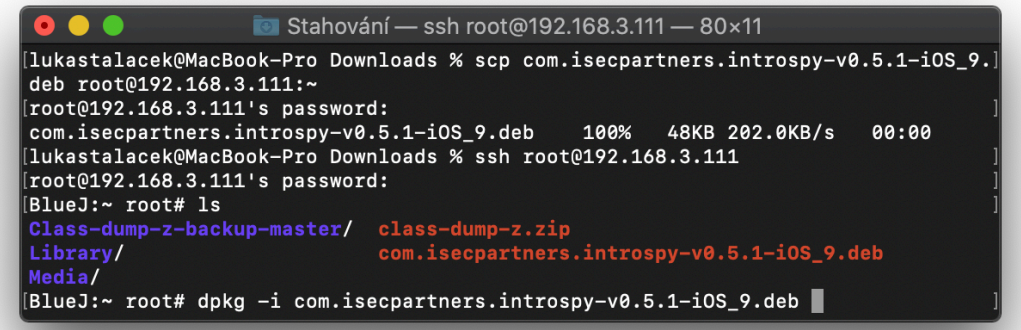
Blackboxový nástroj pro sledování a logování API požadavků, jejich argumentů a návratových hodnot.

Opět je třeba začít stažením z repozitíře na GitHubu. Dále v terminálu pokračovat použitím příkazu pro překopírování do telefonu:

```
⇒ scp -r com.isecpartners.introspy-v0.5.1-iOS_9.deb
  root@192.168.3.111:~
```

Potvrdit heslem *„alpine“* a následně se přihlásit na telefon jako root.

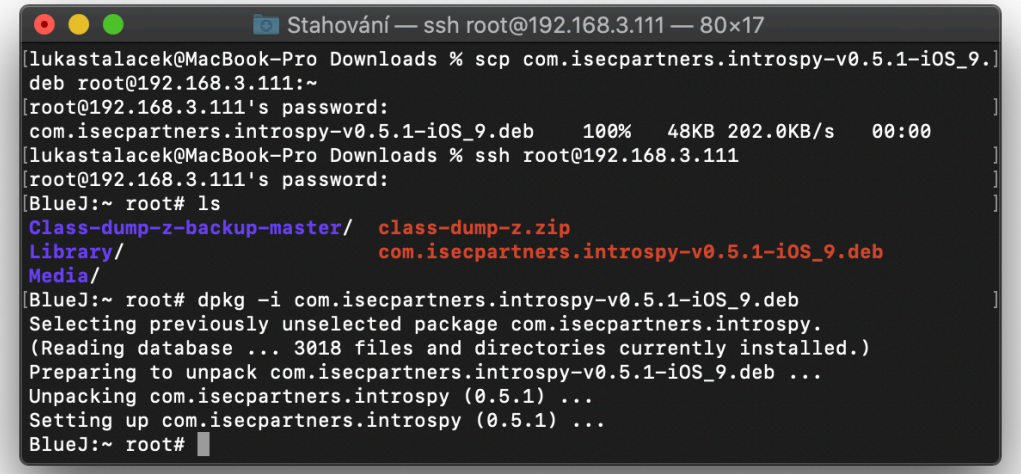
```
⇒ ssh root@192.168.3.111
```



Obrázek 22: Introspy - kopírování

V tomto bodě už zbývá jen použití příkazu *dpkg*:

```
⇒ dpkg -i com.isecpartners.introspy-v0.5.1-iOS_9.deb
```

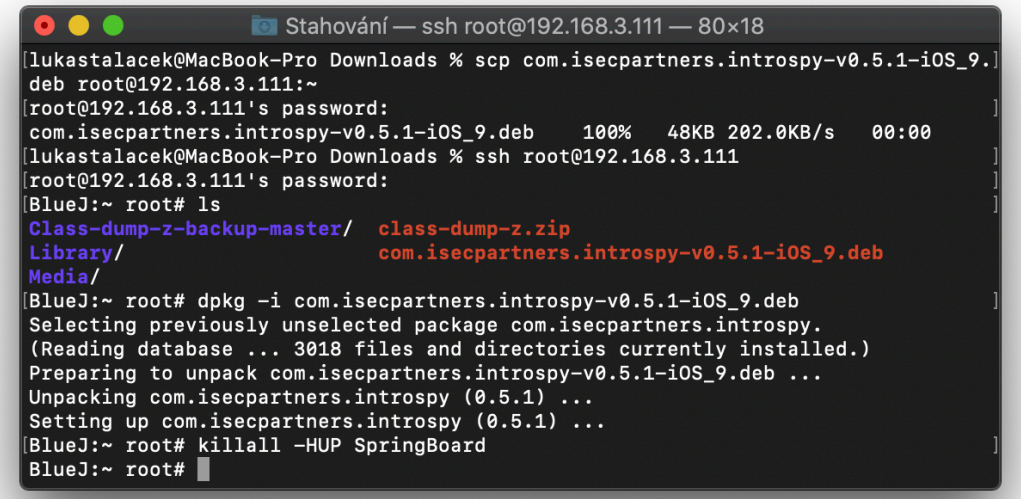


Obrázek 23: Introspy – rozbalení balíčku

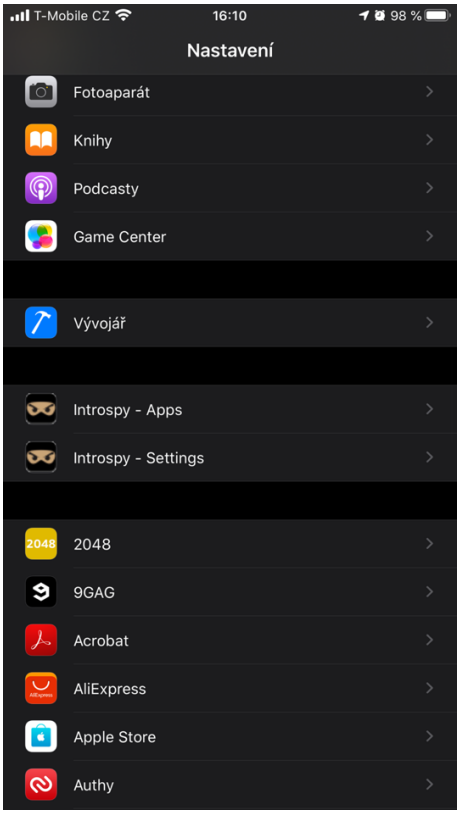
Pro úspěšné zobrazení změn, například v nastavení telefonu, je doporučeno použití příkazu:

```
⇒ killall -HUP SpringBoard
```

Tím se restartuje pouze SpringBoard, ne celý telefon, takže Jailbreak zůstane aktivní.



Obrázek 24: Introspy



Obrázek 25: Nastavení – Introspy

Instalace SSL Kill Switch 2

Blackboxový nástroj pro vypnutí validací certifikátů SSL.

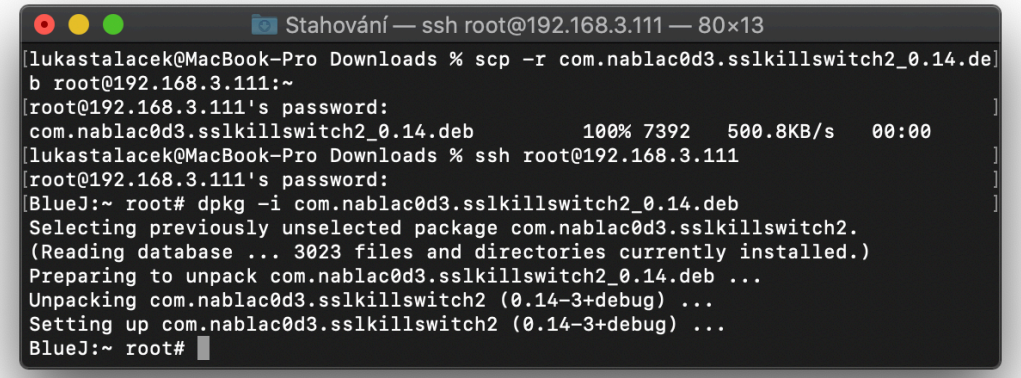
Stejně jako u předešlého nástroje, je nutno stáhnout soubory z repozitáře, opět k nalezení na GitHubu, a následně jej překopírovat do telefonu.

```
⇒ scp -r com.nablac0d3.sslkillswitch2_0.14.deb
  root@192.168.3.111:~
```

Dále se přihlásit do telefonu jako root a použít příkaz dpkg:

```
⇒ ssh root@192.168.3.111
⇒ dpkg -i com.nablac0d3.sslkillswitch2_0.14.deb
```

Doporučeno je také použití příkazu pro respring telefonu.



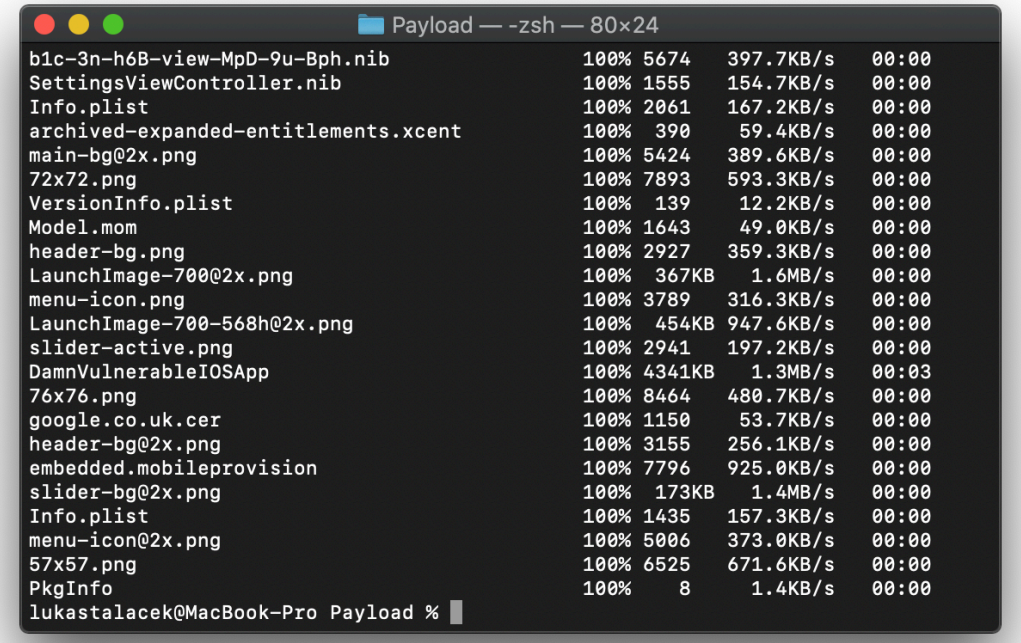
Obrázek 26: SSL Kill Switch 2

Instalace Damn Vulnerable iOS Application (DVIA)

V překladu „Zatraceně zranitelná iOS aplikace“. Poskytuje legální prostředí pro penetrační testování bezpečnostním nadšencům a profesionálům. Tato verze je napsána v jazyce Objective-C a podporuje 32bit i 64bit zařízení.[16]

Z repozitáře je potřeba stáhnout pouze soubor *DamnVulnerableiOSApp.ipa*. Ten za pomocí aplikace *The Unarchiver* rozbalit. Vytvoří se ve stejném umístění rozbalená složka, která obsahuje dvě podsložky – *Payload* a *Symbols*. Důležitá je v tuto chvíli první složka, v té se nachází soubor *DamnVulnerableIOSApp.app* a ten je potřeba překopírovat do telefonu, použitím následujícího příkazu v místě umístění souboru:

```
⇒ scp -r DamnVulnerableIOSApp.app mobile@192.168.3.111:~
```



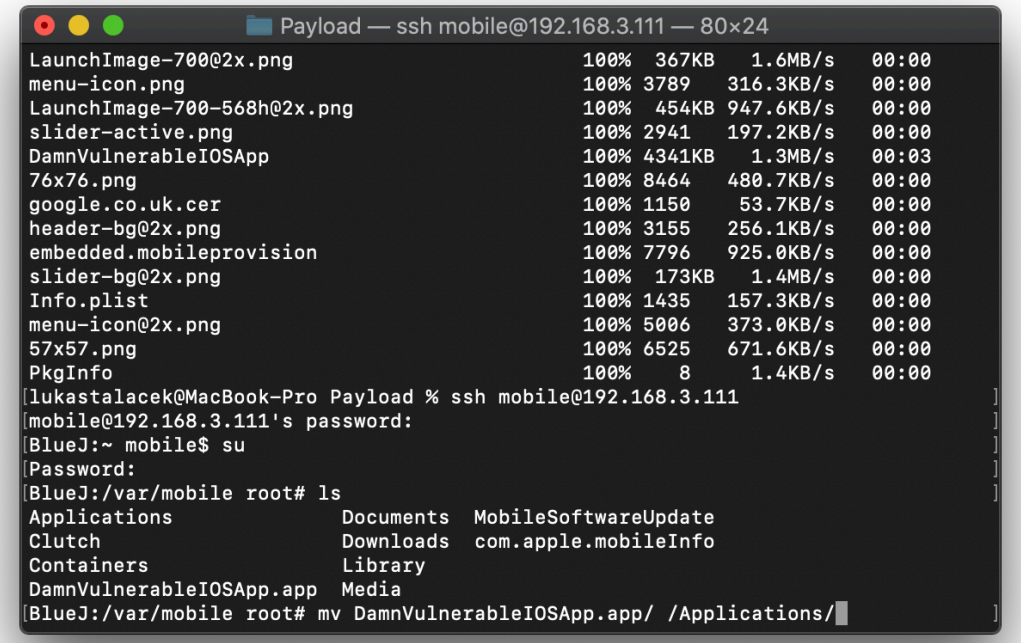
Obrázek 27: Damn Vulnerable iOS App

Po úspěšném kopírování je dalším krokem přihlášení k telefonu.

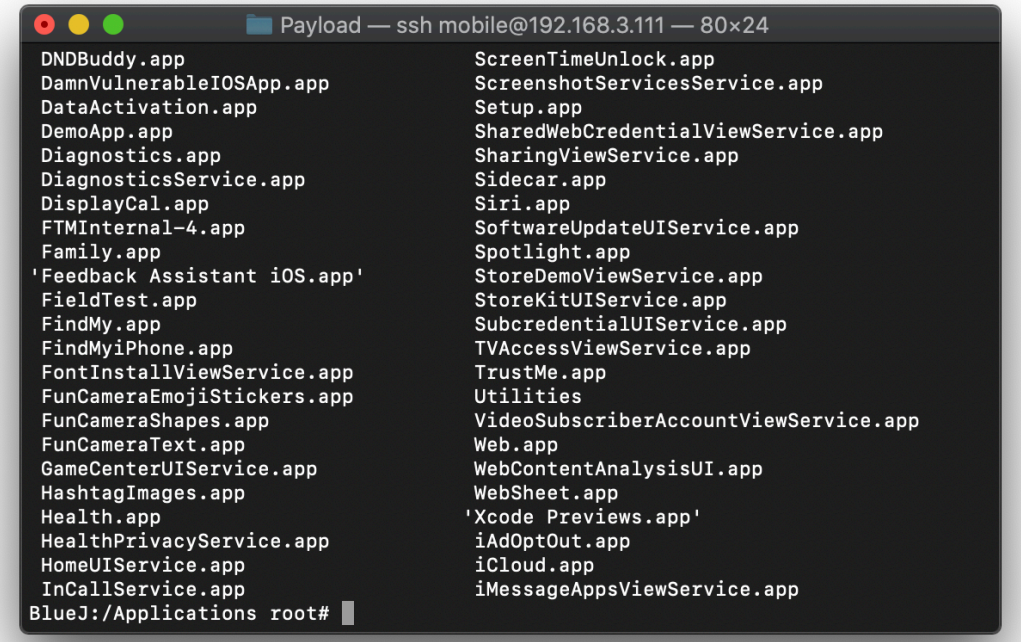
```
⇒ ssh mobile@192.168.3.111
```

Překopírovanou aplikaci je třeba přesunout do složky aplikací, pro tento krok jsou vyžadována root práva, proto je na místě použití příkazu *su*.

```
⇒ su
⇒ mv DamnVulnerableIOSApp.app/ /Applications/
```



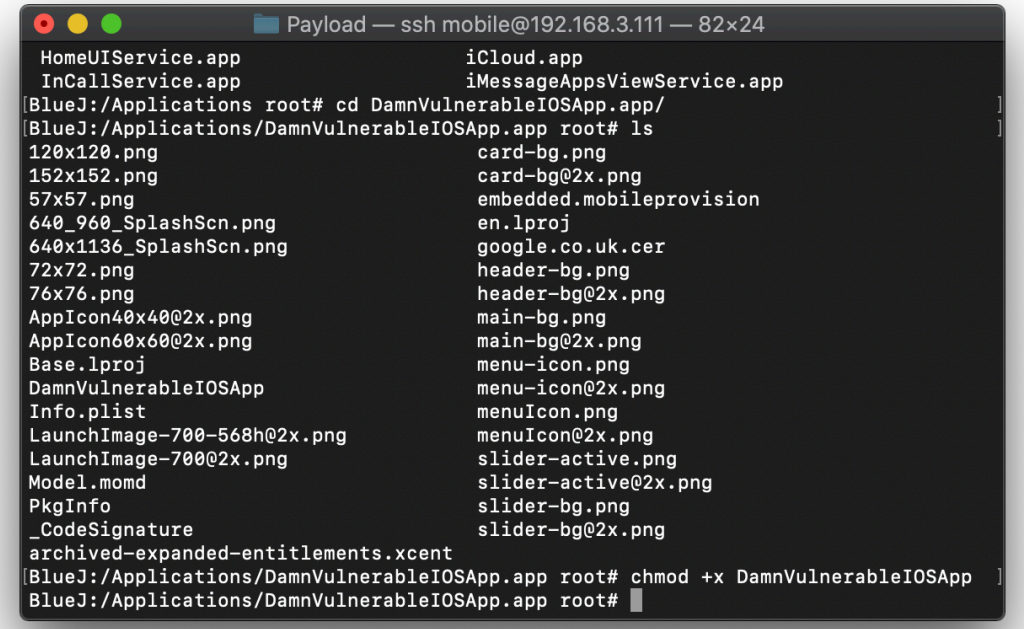
Obrázek 28: Damn Vulnerable iOS App - přesun



Obrázek 29: Damn Vulnerable iOS App – úspěšný přesun

Aby šla aplikace DVIA spustit obyčejnému uživateli, musí se přidat právo pro spuštění. Je nutno vlézt přímo do souborů aplikace:

```
⇒ cd DamnVulnerableIOSApp.app/
⇒ chmod +x DamnVulnerableIOSApp
```



Obrázek 30: Damn Vulnerable iOS App – úprava práv

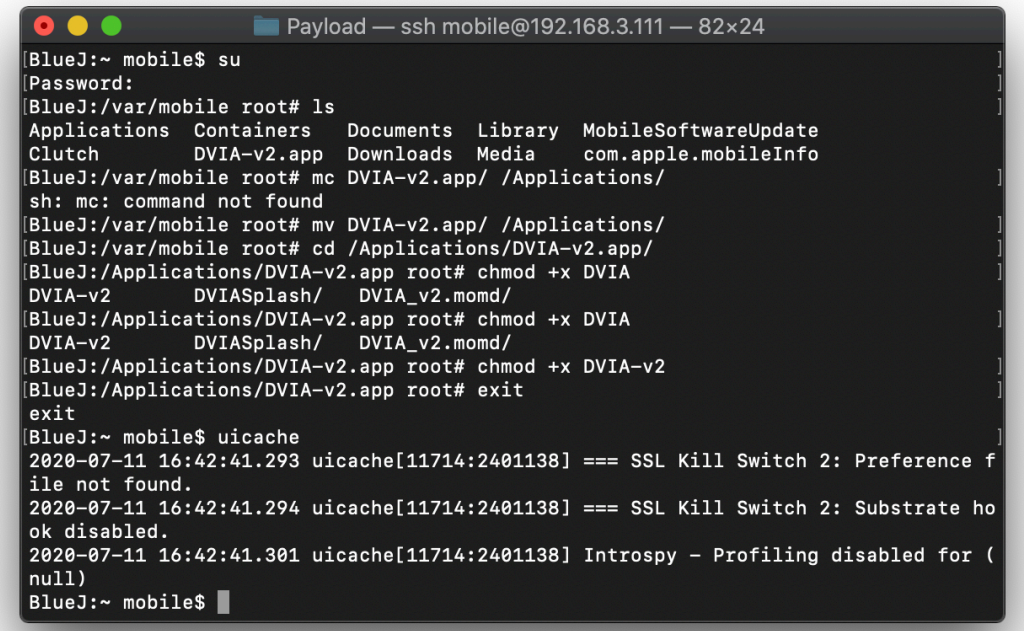
Kvůli dřívějšímu použití příkazu *su* se nyní použije příkaz *exit*, tím se odhlásí uživatel root a aktivní bude opět běžný uživatel – mobile.

Nakonec se použije příkaz *uicache*. Tím se ve zkratce docílí manuálního aktualizování ikon aplikací, které se v případě klasické instalace aplikací oficiální cestou (tedy přes AppStore) provádí samo.

Instalace DVIA 2

Druhá, novější verze „Zatraceně zranitelné iOS aplikace“. Ta už je napsána v jazyce Swift a podporuje pouze 64bit zařízení.

Získání aplikace, i průběh její instalace je naprosto totožný s její předchozí verzí, Liší se pouze v jejím názvu.



Obrázek 31: Damn Vulnerable iOS App 2

Instalace FileExplorer / iFile

Dlouhou dobu byl iFile nejoblíbenějším iOS souborovým průzkumníkem, až donedávna. Od verze iOS 11.2.6 není podporován a spolehlivě jej nahradila aplikace Filza. Jejím účelem je poskytnout uživateli procházet celý systém, jeho složky a soubory v přehledném kabátku a takovou formou, jako jsou uživatelé zvyklí z počítačů. Instalace probíhala přes aplikaci Cydia.

Dokončení přípravy telefonu

Po úspěšné přípravě budou na ploše telefonu nové ikony instalovaných aplikací a telefon je připraven pro testování.

4.4 Shrnutí

Na základě provedeného průzkumu a testování lze konstatovat, že dynamicky se měnící prostředí je komplikací nejen pro vývojáře mobilních aplikací, ale i pro testery. Každý update operačního systému, každý nový model telefonu s sebou nese riziko, že ten, či onen nástroj pro testování nebude fungovat.

V této práci byl sestaven balíček nástrojů, které byly postupně nainstalovány a otestována jejich funkcionalita na aktuální verzi iOS. Lze tedy doporučit následující nástroje:

- Xcode
- Checkra1n
- iExplorer
- Cydia Impactor
- Burp Suite
- SSH
- Clutch
- Introspy iOS
- SSL Kill Switch 2
- DamnVulnerableIOSApp
- DamnVulnerableIOSApp 2
- Filza

ZÁVĚR

Každým rokem vznikají nové společnosti zaměřené na vývoj software. Každým rokem také vznikají nové projekty. Vývoj takových projektů, ať už malých, nebo i těch větších, stojí určité finance a čas. Pokud se něco nepovede na první dobrou, a to rozhodně nikdy nebude, je třeba to potom složitě napravovat, a to znamená další čas, který se tomu musí vývojáři věnovat. Musejí chybu nalézt, následně opravit. Kvůli tomu se vývoj může znatelně prodražit a v případě, že chyba bude souviset s bezpečností uživatelů a jejich dat, finance nebudou jediný problém, byla by to hotová katastrofa. Aby se předešlo takovým omylům, je velice vhodné software testovat.

V dnešní době má mobilní telefon už i dokonce dítě. Tyto telefony neboli smartphony, umožňují přístup na internet, a nejen mládež zde tráví mnoho času. Nekontrolovatelně prochází web, stahují hry, využívají aplikace sociálních sítí. Sdílí osobní a bankovní informace, své i svých rodičů. Telefon je s nimi naprosto všude, s telefonem žijí.

Pro příklad na telefonu není uživatel informován o aktivním využívání fotoaparátu. Velmi nepříjemné, že ano? Nad tím je třeba hluboce se zamyslet.

S rostoucí popularitou internetu a s ním i souvisejících aktivit rostou počty osob, které jsou hrozbou pro obyčejné lidi. Osoby, které své znalosti využívají ke krádežím citlivých informací. Informací o kreditních kartách. Zjišťují informace, kterými následně obyčejné lidi, děti, zneužívají, vydírají.

Cílem této práce bylo zjistit, jaké jsou možnosti testování aplikací na platformě iOS. Zjistit, jaký software je k tomu vhodný a jak jej získat. Cílem bylo, aby i neprofesionál z oboru nabyl alespoň základních znalostí v tomto směru a minimálně začal být obezřetný při své aktivitě na internetu.

V průběhu druhé části práce je uveden postup pro instalaci dvou aplikací, které umožňují nahlédnout do bezpečnostních skulinek. Velmi doporučuji prozkoumat je hlouběji.

SEZNAM POUŽITÉ LITERATURY

[1] VÁVRŮ, Jiří. *IPhone: vývoj aplikací*. Praha: Grada, 2012, 179 s. Průvodce. ISBN 9788024744575.

[2] HEROUT, Pavel. *Testování pro programátory*. České Budějovice: Kopp, 2016, 405 s. ISBN 9788072324811.

[3] BUREŠ, Miroslav, Miroslav RENDA, Michal DOLEŽEL, Peter SVOBODA, Zdeněk GRÖSSL, Martin KOMÁREK, Ondřej MACEK a Radoslav MLYNÁŘ. *Efektivní testování softwaru: klíčové otázky pro efektivitu testovacího procesu*. Praha: Grada, 2016, 229 s. Profes!onal. ISBN 9788024755946.

[4] MEERTS, Joris a Dorothy GRAHAM. The History of Software Testing. *Testing References* [online]. c2010-2019 [cit. 2020-07-30]. Dostupné z: http://www.testingreferences.com/testinghistory.php

[5] KITNER, Radek. Co je testování softwaru? *Kitner* [online]. c2015 [cit. 2020-07-30]. Dostupné z: https://kitner.cz/testovani_softwaru/co-je-testovani-softwaru/

[6] Testování softwaru - co to je? *Testování software* [online]. [cit. 2020-07-30]. Dostupné z: http://test.swtestovani.cz/index.php?option=com_content&view=article&id=7:co-to-je-to-testovani&catid=3:zaklady&Itemid=11

[7] *Checkra1n* [online]. Kim Jong Cracks, c2019-2020 [cit. 2020-07-19]. Dostupné z: https://checkra.in/

[8] , Rupesh. IOS Layered Architecture. *IOS Programming Tricks: iOS Layered Architecture* [online]. 2017 [cit. 2020-07-30]. Dostupné z: https://codeingwithios.blogspot.com/2017/09/ios-layered-architecture.html

[9] CASTRO, Kristi. How are iOS and Android similar? How are they different? *Tutorialspoint* [online]. 2018 [cit. 2020-07-30]. Dostupné z: https://www.tutorialspoint.com/how-are-ios-and-android-similar-how-are-they-different

[10] ĎURKECH, Tomáš. Velké srovnání nejpopulárnějších operačních systémů Android a iOS. Který je lepší? *Refresher* [online]. 2015 [cit. 2020-07-30]. Dostupné z: https://refresher.cz/29650-Velke-srovnani-nejpopularnejsich-operacnich-systemu-Android-a-iOS-Ktery-je-lepsi?gdpr-accept=1

[11] HOLZMAN, Ondřej. Recovery a DFU režim: jak na ně a jaký je v nich rozdíl. *Jablíčkář* [online]. 2012 [cit. 2020-07-30]. Dostupné z: https://jablickar.cz/recovery-a-dfu-rezim-jak-na-ne-a-jaky-je-v-nich-rozdil/

[12] GIANCHANDANI, Prateek. IOS Application security: Part 1 – Setting up a mobile pentesting platform. *Infosec* [online]. 2013 [cit. 2020-07-30]. Dostupné z: https://resources.infosecinstitute.com/ios-application-security-part-1-setting-up-a-mobile-pentesting-platform/#gref

[13] , Sven. IOS Basic Security Testing. *Mobile Security Testing Guide* [online]. [cit. 2020-07-30]. Dostupné z: https://mobile-security.gitbook.io/mobile-security-testing-guide/ios-testing-guide/0x06b-basic-security-testing

[14] BOUCHARD, Anthony. Understanding untethered, semi-untethered, semi-tethered, and tethered jailbreaks. *IDownloadBlog* [online]. 2019 [cit. 2020-07-30]. Dostupné z: https://www.idownloadblog.com/2019/11/21/types-of-jailbreaks/

[15] Untethered Jailbreak vs Semi-Untethered Jailbreak vs Tethered Jailbreak. *IPhoneHeat* [online]. 2011 [cit. 2020-07-30]. Dostupné z: http://www.iphoneheat.com/2011/10/untethered-tethered-semi-untethered-jailbreak/

[16] *Damn Vulnerable iOS Application (DVIA)* [online]. c2020 [cit. 2020-08-04]. Dostupné z: http://damnvulnerableiosapp.com/

SEZNAM OBRÁZKŮ

Obrázek 1: Typy testování 12
Obrázek 2: Vrstvy iOS 15
Obrázek 3: Xcode 23
Obrázek 4: Simulator 24
Obrázek 5: Instalace checkra1n 25
Obrázek 6: Spuštění nástroje checkra1n 26
Obrázek 7: Aplikace checkra1n (vlevo) a instalace aplikace Cydia 27
Obrázek 8: iExplorer 29
Obrázek 9: Cydia Impactor 30
Obrázek 10: Burp Suite - volba projektu 31
Obrázek 11: Burp Suite - volba konfig. souboru 32
Obrázek 12: Burp Suite - Hlavní okno 32
Obrázek 13: Burp Suite - Záložka proxy 33
Obrázek 14: Burp Suite - Nastavení proxy 33
Obrázek 15: konfigurace proxy na telefonu (vlevo) a ověření připojení k Burp Suite 34
Obrázek 16: Burp Suite - Stažení certifikátu (vlevo) a instalace na iOS 35
Obrázek 17: Burp Suite - Odposlouchávání proxy 36
Obrázek 18: Burp Suite - Event log 36
Obrázek 19: SSH 37
Obrázek 20: Clutch – build 39
Obrázek 21: Clutch 40
Obrázek 22: Introspy - kopírování 41
Obrázek 23: Introspy – rozbalení balíčku 41
Obrázek 24: Introspy 42
Obrázek 25: Nastavení – Introspy 42
Obrázek 26: SSL Kill Switch 2 43
Obrázek 27: Damn Vulnerable iOS App 44
Obrázek 28: Damn Vulnerable iOS App - přesun 45
Obrázek 29: Damn Vulnerable iOS App – úspěšný přesun 45
Obrázek 30: Damn Vulnerable iOS App – úprava práv 46
Obrázek 31: Damn Vulnerable iOS App 2 47

Seznam tabulek

Tabulka 1: Rozdíly mezi iOS a Android 19